



TUGAS AKHIR - TF 181801

**ANALISIS PERFORMANSI SISTEM PENGENDALIAN PADA
KOLOM DISTILASI MENGGUNAKAN FUZZY-WHALE
OPTIMIZATION ALGORITHM (FUZZY-WOA)**

AGAM PRIHATNA
NRP. 02311640000027

Dosen Pembimbing:
Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.

Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya
2020



TUGAS AKHIR - TF 181801

**ANALISIS PERFORMANSI SISTEM PENGENDALIAN PADA
KOLOM DISTILASI MENGGUNAKAN FUZZY-WHALE
OPTIMIZATION ALGORITHM (FUZZY-WOA)**

AGAM PRIHATNA
NRP. 02311640000027

Dosen Pembimbing:
Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.

Departemen Teknik Fisika
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya
2020

Halaman ini sengaja dikosongkan



FINAL PROJECT - TF 181801

***PERFORMANCE ANALYSIS OF FUZZY-WOA BASED
DISTILLATION COLUMN CONTROL***

**AGAM PRIHATNA
NRP. 02311640000027**

**Supervisors:
Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.**

***Department of Engineering Physics
Faculty of Industrial Technology and System Engineering
Institut Teknologi Sepuluh Nopember
Surabaya
2020***

Halaman ini sengaja dikosongkan

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini.

Nama : Agam Prihatna
NRP : 02311640000027
Departemen / Prodi : Teknik Fisika / S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa tugas akhir dengan judul "*Analisis Performansi Sistem Pengendalian pada Kolom Distilasi Menggunakan Fuzzy-Whale Optimization Algorithm (Fuzzy-WOA)*" adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada tugas akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Pamekasan, 4 Agustus 2020

Yang membuat pernyataan,



Agam Prihatna

NRP. 02311640000027

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN - 1
TUGAS AKHIR

**ANALISIS PERFORMANSI SISTEM PENGENDALIAN PADA
KOLOM DISTILASI MENGGUNAKAN FUZZY-WHALE
OPTIMIZATION ALGORITHM (FUZZY-WOA)**

Oleh:


Agam Prihatna
NRP. 02311640000027

Surabaya, 4 Agustus 2020


Menyetujui,
Pembimbing I


Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
NIP. 19660116 198903 2 001

Menyetujui,
Pembimbing II


Dr. Ir. Svamsul Arifin, M.T.
NIP. 19630907 198903 1 004

Mengetahui,
Kepala Departemen

Teknik Fisika FTIRS-ITS

Dr. Suvanto S.T., M.T.
DEPARTEMEN
FISIKA
NIP. 19741113 199512 1 002

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN - 2

ANALISIS PERFORMANSI SISTEM PENGENDALIAN PADA KOLOM DISTILASI MENGGUNAKAN FUZZY-WHALE OPTIMIZATION ALGORITHM (FUZZY-WOA)

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Progam Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Institut Teknologi Sepuluh Nopember


Oleh:

AGAM PRIHATNA

NRP. 02311640000027

Disetujui oleh Tim Penguji Tugas Akhir:

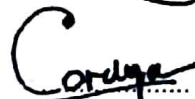
1. Prof. Dr. Ir. Aulia Siti Aisjah, M.T.

 (Pembimbing I)

2. Dr. Ir. Syamsul Arifin, M.T.

 (Pembimbing II)


3. Hendra Cordova, S.T., M.T.

 (Ketua Penguji)

4. Ir. Harsono, M.Sc., Ph.D.

 (Penguji I)

5. Dr.rer.nat. Ruri Agung W., S.T., M.T.

 (Penguji II)

SURABAYA

2020

Halaman ini sengaja dikosongkan

ANALISIS PERFORMANSI SISTEM PENGENDALIAN PADA KOLOM DISTILASI MENGGUNAKAN FUZZY-WHALE OPTIMIZATION ALGORITHM (FUZZY-WOA)

Nama : Agam Prihatna
NRP : 02311640000027
Departemen : Teknik Fisika FTIRS – ITS
Dosen Pembimbing : Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.

ABSTRAK

Sistem pengendalian komposisi pada sebuah kolom distilasi, selama ini masih banyak menggunakan pengendali konvensional PID dan pengendali fuzzy tanpa optimasi. Hasil performansi yang didapatkan oleh kedua pengendali tersebut masih belum menghasilkan hasil performansi yang optimal, sehingga dilakukanlah penelitian mengenai sistem pengendalian komposisi distilat dan komposisi bawah pada kolom distilasi menggunakan *fuzzy-whale optimization algorithm* (fuzzy-WOA) untuk mendapatkan hasil performansi yang lebih optimal. Jenis kolom distilasi yang dikendalikan pada penelitian ini adalah kolom distilasi biner methanol-air Wood & Berry. Penelitian ini terdiri dari tiga metode pengendali, yaitu pengendali PID dengan tiga tipe rancangan (P, PI, PID), pengendali fuzzy tanpa optimasi dengan dua skenario rancangan, dan pengendali fuzzy-WOA dengan empat skenario rancangan. Skenario pada pengendali fuzzy tanpa optimasi memiliki variasi pada bentuk fungsi keanggotaan, sedangkan untuk skenario pengendali fuzzy-WOA memiliki variasi pada parameter WOA yaitu berupa jumlah *search agent* dan ini iterasi maksimum. Simulasi dilakukan dengan menggunakan *simulink* Matlab untuk mendapatkan hasil respons ketiga pengendali. Hasil respons antar ketiga pengendali dari tipe dan skenario terbaiknya dibandingkan untuk menentukan pengendali mana yang menghasilkan performansi terbaik. Hasil yang didapat menyimpulkan bahwasanya pengendali *fuzzy-whale optimization algorithm* mampu menghasilkan performansi yang lebih baik daripada pengendali

konvensional PID dan pengendali fuzzy tanpa optimasi. Didapat nilai performansi pada komposisi distilat yaitu *rise time*, *settling time*, *maximum overshoot*, dan *error steady state* berturut-turut sebesar 19,47 detik, 31,86 detik, 1,55%, dan 0,98% serta pada komposisi bawah berturut-turut sebesar 0,22 detik, 74,30 detik, 1328%, dan 0,30%.

Kata Kunci: kolom distilasi, logika fuzzy, PID, *whale optimization algorithm*

PERFORMANCE ANALYSIS OF FUZZY-WOA BASED DISTILLATION COLUMN CONTROL

Name : Agam Prihatna
NRP : 02311640000027
Department : Engineering Physics FTIRS - ITS
Supervisors : Prof. Dr. Ir. Aulia Siti Aisjah, M.T.
Dr. Ir. Syamsul Arifin, M.T.

ABSTRACT

A composition control system in distillation column plant typically still uses conventional PID and/or fuzzy controller without optimization. The performance results obtained by such controllers do not produce optimal performance results, so a study on the distillation composition control system in the distillation column using fuzzy-whale optimization algorithm (fuzzy-WOA) is conducted to get more optimal performance results. The type of distillation column controlled in this study is Wood & Berry's methanol-air binary distillation column. This study consists of three control methods, namely PID controller with three design types (P, PI, PID), fuzzy controller without optimization with two design scenarios, and fuzzy-WOA controller with four design scenarios. Scenarios for fuzzy controllers without optimization vary in the form of membership functions, whereas for scenarios of fuzzy-WOA controllers we have variations in WOA parameters, namely the number of search agents and the maximum number of iterations. The simulation is done by using Matlab simulink to get the response of the three controllers. The best response results among the three controllers of any scenario are compared to determine which controller produces the best performance. The results obtained concluded that the fuzzy-whale optimization algorithm controller is able to produce better performance than conventional PID controllers and fuzzy controllers without optimization. The performance values obtained for the distillate composition are rise time, settling time, maximum overshoot, and steady state error, which are 19.47

seconds, 31.86 seconds, 1.55%, and 0.98% respectively and the bottom composition respectively also contributed by 0.22 seconds, 74.30 seconds, 1328% and 0.30% respectively.

Keywords: distillation column, fuzzy logic, PID, whale optimization algorithm

KATA PENGANTAR

Puji syukur kehadiran Allah SWT. yang telah melimpahkan rahmat serta hidayah-Nya sehingga laporan tugas akhir yang berjudul “**Analisis Performansi Sistem Pengendalian pada Kolom Distilasi Menggunakan Fuzzy-Whale Optimization Algorithm (Fuzzy-WOA)**” dapat diselesaikan dengan baik dan tepat waktu. Selama proses pengerjaan tugas akhir, terdapat banyak pihak yang turut membantu baik dalam segi pengetahuan maupun dalam segi psikologis. Oleh karena itu, ucapan terimakasih disampaikan kepada pihak-pihak yang telah membantu selama proses pengerjaan tugas akhir ini, diantaranya:

1. Keluarga besar Drs. Eka Judyta Setiawan yang senantiasa memberikan dukungan dan fasilitas selama proses pengerjaan tugas akhir.
2. Diri sendiri yang selalu berusaha bangkit untuk menyelesaikan tugas akhir yang sangat sulit ini.
3. Dr. Suyanto, S.T., M.T. selaku Kepala Departemen Teknik Fisika ITS yang telah bertanggung jawab atas segala hal terkait tugas akhir mahasiswa Teknik Fisika ITS.
4. Dr. Katherin Indriawati, S.T., M.T. selaku Sekretaris Departemen Teknik Fisika ITS yang selalu siap siaga membantu mahasiswa selama proses pengerjaan tugas akhir.
5. Prof. Dr. Ir. Aulia Siti Aisjah, M.T. dan Dr. Ir. Syamsul Arifin, M.T. selaku pembimbing tugas akhir yang telah memberikan saran dan masukan selama proses pengerjaan tugas akhir.
6. Totok Ruki Biyanto, S.T., M.T., Ph.D. dan Hendra Cordova, S.T., M.T. selaku dosen Bidang Minat Laboratorium Rekayasa Instrumentasi, Kontrol, dan Optimasi yang telah memberikan ilmu dan saran dalam pengerjaan tugas akhir.
7. Erna Septyaningrum, S.T., M.T. yang telah membantu mencari proposal tugas akhir penulis saat laptop penulis sedang mengalami kerusakan.
8. Raihan Sergi Perdana, Kartini, Rizqi Lazuardy Manararrahmah, Novita Amalia, Yunita Puspita Rini, Yoanne Ayuningtyas Sania, Barakatul Khairat, Tysna Mayang Hapsari, Mohamad Faizul Fikri, Dea Faiza Febrianty, Addien

Wicaksono, Wisnu Wira Dharma, Annisa Jasmine Rudend, Maharani Sarah Holle, Bariz Irfan Ridho, Amrizal Tariq Chalidi, Kurnia Diastana Abdirandra, dan Muhammad Ja'far Shodiq selaku sahabat yang selalu menjadi tempat berkeluh-kesah dan selalu memberikan motivasi, bantuan, serta semangat selama proses pengerjaan tugas akhir.

9. Viko Dian Nano, Ahmad Auril Barelvi, Ian Haikal Amir Akbar, Muhammad Mukhlis Jamali, dan Achmad Ferry Bastian selaku teman sejawat Kolom Distilasi dan Kapal yang telah membantu banyak hal selama proses pengerjaan tugas akhir.
10. Kevin Gica Erwandha, Haryo Arif Wicaksono, Irvandhi Hito Hageshi, Bella Widya Gunawan, dan Moh Danni Aditya selaku kakak tingkat di Laboratorium Pengukuran, Keandalan, Risiko, dan Keselamatan serta Yuda Pambudi selaku kakak tingkat di PSM yang telah memberikan banyak ilmu dalam menunjang selesainya laporan tugas akhir ini.
11. Vedanta Agam, Noviarizqoh Nurul Habibah, Anggun Cahyaning Lintang Respati, Dwi Maulida Zakkiyah, Hijriyatul Fajriyah, Made Widnyana Andika, Muchamad Arief Dharmawan, M. Hilmi, Shiang Ni Putri Mentari, Roby Bagaskara Bahasyar, dan Muhammad Arif Samudya selaku adik tingkat di Laboratorium Pengukuran, Keandalan, Risiko, dan Keselamatan yang selalu memberikan keceriaan.
12. Semua pihak yang turut membantu selama proses pengerjaan tugas akhir yang tidak bisa disebutkan satu-persatu.

Akhir kata, semoga laporan tugas akhir ini dapat memberikan banyak manfaat, informasi, dan juga pengetahuan bagi para pembaca. Permintaan maaf juga tidak lupa disampaikan bagi para pembaca apabila terdapat kekurangan pada laporan tugas akhir ini, serta diharapkan bagi para pembaca untuk dapat mengembangkan tugas akhir ini.

Pamekasan, 4 Agustus 2020

Agam Prihatna

DAFTAR ISI

HALAMAN JUDUL.....	i
COVER PAGE.....	iii
PERNYATAAN BEBAS PLAGIASI.....	v
LEMBAR PENGESAHAN - 1	vii
LEMBAR PENGESAHAN - 2	ix
ABSTRAK	xi
ABSTRACT.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
DAFTAR NOTASI.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	4
1.5 Sistematika Laporan	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2 Kolom Distilasi Biner.....	7
2.3 Model <i>First Order Plus Death Time</i> FOPDT.....	14
2.4 Sistem Pengendalian <i>Multiple Input Multiple Output</i> 2x2 (MIMO)	15
2.5 Karakteristik Respons Sistem.....	17

2.6	Pengendali Konvensional <i>Proportional-Integral-Derivative</i> (PID)	20
2.7	Sistem Logika Fuzzy.....	24
2.8	<i>Mean Squared Error</i> (MSE)	34
2.9	<i>Whale Optimization Algorithm</i> (WOA)	35
BAB III METODOLOGI PENELITIAN		43
3.1	Pengumpulan Data Proses pada Kolom Distilasi.....	44
3.2	Pemodelan Matematis dan Uji <i>Open Loop</i> Kolom Distilasi.....	46
3.3	Perancangan Sistem Pengendali	48
BAB IV HASIL DAN PEMBAHASAN		71
4.1	Hasil Uji Respons <i>Step</i> Sistem <i>Open Loop</i>	71
4.2	Hasil Simulasi Pengendali PID	72
4.3	Hasil Simulasi Pengendali Fuzzy.....	77
4.4	Hasil Simulasi Pengendali Fuzzy-WOA.....	80
4.5	Perbandingan Hasil Simulasi Ketiga Metode Pengendali	84
BAB V KESIMPULAN DAN SARAN		89
5.1	Kesimpulan	89
5.2	Saran	89
DAFTAR PUSTAKA.....		91
LAMPIRAN		95
BIODATA PENULIS		115

DAFTAR GAMBAR

Gambar 2.1 Kolom Distilasi Biner	8
Gambar 2.2 Ilustrasi Operasi Kolom Distilasi.....	10
Gambar 2.3 Tangki Akumulator dan Kondenser	12
Gambar 2.4 Bagian Bawah Kolom Distilasi	13
Gambar 2.5 Struktur Kontrol L-V Kolom Distilasi.....	14
Gambar 2.6 Interaksi Sistem MIMO 2x2	15
Gambar 2.7 Diagram Blok Sistem Kendali MIMO 2x2 Terdesentralisasi.....	16
Gambar 2.8 Karakteristik Respons Sistem Orde-2.....	18
Gambar 2.9 Blok Diagram Pengendali PID	23
Gambar 2.10 Diagram Blok Sistem Logika Fuzzy.....	26
Gambar 2.11 Representasi Kurva Segitiga.....	27
Gambar 2.12 Representasi Kurva Trapesium.....	28
Gambar 2.13 Representasi Kurva Bahu	29
Gambar 2.14 Teknik <i>Bubble-net</i> Paus Bungkok	35
Gambar 2.15 (a) Posisi Vektor 2D (b) Posisi Vektor 3D dan Kemungkinan Posisi yang Akan Datang.....	37
Gambar 2.16 (a) <i>Shrinking Encircling Mechanism</i> (b) <i>Spiral Updating Position</i>	38
Gambar 2.17 Fase Eksplorasi pada WOA	40
Gambar 2.18 <i>Flowchart</i> Algoritma WOA.....	41
Gambar 3.1 Diagram Alir Penelitian.....	44
Gambar 3.2 Blok Diagram dari <i>Plant</i> Kolom Distilasi dengan Dua Variabel <i>Input</i>	46
Gambar 3.3 Uji Respons <i>Step</i> Sistem <i>Open Loop</i>	48
Gambar 3.4 Diagram Blok Sistem Pengendalian MIMO Desentralisasi 2X2	50
Gambar 3.5 Diagram Blok Sistem Pengendalian Komposisi pada Kolom Distilasi dengan Pengendali PID	51
Gambar 3.6 Simulasi Pengendali PID pada <i>Simulink</i>	53

Gambar 3.7 Diagram Blok Sistem Pengendalian Komposisi pada Kolom Distilasi dengan Pengendali Fuzzy tanpa Optimasi	54
Gambar 3.8 <i>Input Error</i> dengan Fungsi Keanggotaan Bentuk Bahu	56
Gambar 3.9 <i>Input Error</i> dengan Fungsi Keanggotaan Bentuk Segitiga	56
Gambar 3.10 <i>Input Delta Error</i> dengan Fungsi Keanggotaan Bentuk Bahu	57
Gambar 3.11 <i>Input Delta Error</i> dengan Fungsi Keanggotaan Bentuk Segitiga ..	57
Gambar 3.12 <i>Output</i> Sistem dengan Fungsi Keanggotaan Bentuk Bahu	58
Gambar 3.13 <i>Output</i> Sistem dengan Fungsi Keanggotaan Bentuk Segitiga	58
Gambar 3.14 Simulasi Pengendali Fuzzy pada <i>Simulink</i>	60
Gambar 3.15 Diagram Alir Optimasi Parameter <i>Fuzzy Logic Controller</i>	62
Gambar 3.16 <i>Input Error</i> Hasil Optimasi pada Skenario 1	67
Gambar 3.17 <i>Input Delta Error</i> Hasil Optimasi pada Skenario 1	67
Gambar 3.18 <i>Output</i> Sinyal Kontrol Hasil Optimasi pada Skenario 1	68
Gambar 3.19 Diagram Blok Sistem Pengendalian Komposisi pada Kolom Distilasi dengan Pengendali Fuzzy-WOA	69
Gambar 4.1 Respons <i>Step</i> Sistem <i>Open Loop</i> (a) Komposisi Distilat terhadap <i>Flow</i> Refluks (b) Komposisi Distilat terhadap <i>Flow Steam</i> (c) Komposisi Bawah terhadap <i>Flow</i> Refluks (d) Komposisi Bawah terhadap <i>Flow Steam</i>	71
Gambar 4.2 Grafik Respons Pengendali PID Komposisi Distilat	73
Gambar 4.3 Grafik Respons Pengendali PID Komposisi Bawah	73
Gambar 4.4 Grafik Respons Pengendali Fuzzy Komposisi Distilat	77
Gambar 4.5 Grafik Respons Pengendali Fuzzy Komposisi Bawah	78
Gambar 4.6 Grafik Respons Pengendali Fuzzy-WOA Komposisi Distilat	81
Gambar 4.7 Grafik Respons Pengendali Fuzzy-WOA Komposisi Bawah	81
Gambar 4.8 Grafik Respons Komposisi Distilat Ketiga Metode	85
Gambar 4.9 Grafik Respons Komposisi Bawah Ketiga Metode	85

DAFTAR TABEL

Tabel 2.1 Pengaruh PID terhadap Performansi Sistem	22
Tabel 3.1 Data <i>Flow</i> dan Komposisi pada Kondisi <i>Steady</i>	45
Tabel 3.2 Data Suhu pada Kondisi <i>Steady</i>	45
Tabel 3.3 Analisis Pasangan <i>Input</i> dan <i>Output</i>	49
Tabel 3.4 Parameter PID pada Pengendali 1	52
Tabel 3.5 Parameter PID pada Pengendali 2	52
Tabel 3.6 Skenario Sistem Pengendali Fuzzy	55
Tabel 3.7 Tabel <i>Rule Base</i> Sistem	59
Tabel 3.8 Skenario Perancangan Fuzzy-WOA.....	61
Tabel 3.9 Hasil Optimasi Parameter Fuzzy pada Variabel <i>Input Error</i>	64
Tabel 3.10 Hasil Optimasi Parameter Fuzzy pada Variabel <i>Input Delta Error</i> ...	65
Tabel 3.11 Hasil Optimasi Parameter Fuzzy pada Variabel <i>Output</i> Sinyal Kontrol	66
Tabel 4.1 Performansi Pengendali PID Komposisi Distilat	74
Tabel 4.2 Performansi Pengendali PID Komposisi Bawah.....	74
Tabel 4.3 Performansi Pengendali Fuzzy Komposisi Distilat.....	79
Tabel 4.4 Performansi Pengendali Fuzzy Komposisi Bawah	79
Tabel 4.5 Performansi Pengendali Fuzzy-WOA Komposisi Distilat.....	82
Tabel 4.6 Performansi Pengendali Fuzzy-WOA Komposisi Bawah.....	82
Tabel 4.7 Performansi Ketiga Metode Pengendali Komposisi Distilat Terbaik ..	86
Tabel 4.8 Performansi Ketiga Metode Pengendali Komposisi Bawah Terbaik...	86

Halaman ini sengaja dikosongkan

DAFTAR NOTASI

F	laju aliran <i>feed</i>	kg/s
D	laju aliran <i>distillate product</i>	kg/s
B	laju aliran <i>bottom product</i>	kg/s
x_F	fraksi <i>feed</i>	mol
x_D	fraksi <i>distillate product</i>	mol
x_B	fraksi <i>bottom product</i>	mol
h_F	entalpi pada <i>feed</i>	J/kg
h_D	entalpi pada <i>distillate</i>	J/kg
h_B	entalpi pada <i>bottom</i>	J/kg
Q_r	<i>heatflow reboiler</i>	J/kg
Q_c	<i>heatflow</i> kondenser	J/kg
M_D	total massa yang tertahan	kg
V_{NT}	laju aliran massa uap	kg/s
L_{NT+1}	laju aliran <i>reflux</i>	kg/s
y_{NT}	fraksi dari uap keluaran kolom distilasi	mol
H_{NT}	entalpi pada uap keluaran <i>top stage</i> kolom	J/kg
h_{NT+1}	entalpi pada <i>reflux</i>	J/kg
L_1	laju aliran keluaran <i>bottom stage</i> kolom	kg/s
V_{RB}	laju aliran uap dari <i>reboiler</i>	kg/s
x_1	fraksi keluaran <i>bottom stage</i> kolom	mol
y_B	fraksi uap dari <i>reboiler</i>	mol
h_1	entalpi keluaran <i>bottom stage</i> kolom	J/kg
H_B	entalpi uap dari <i>reboiler</i>	J/kg
K	<i>gain</i> statis proses	-
τ	konstanta waktu tunak	detik
θ	<i>death time</i>	detik
ξ	rasio redaman	-
ω_n	frekuensi natural	rad/s
ω_n	frekuensi redaman	rad/s

K_p	penguatan <i>proportional</i>	-
K_i	penguatan <i>integral</i>	-
K_d	penguatan <i>derivative</i>	-
T_i	waktu <i>integral</i>	detik
T_d	waktu <i>derivative</i>	detik
WA	nilai rata-rata	-
a_n	nilai predikat aturan ke-n	-
z_n	indeks nilai <i>output</i> (konstanta) ke-n	-
x_i	nilai data asli amatan ke-i	-
F_i	nilai ramalan amatan ke-i	-
n	banyaknya data	-
t	iterasi saat ini	-
\vec{A}	vektor koefisien	-
\vec{C}	vektor koefisien	-
\vec{X}^*	vektor posisi dari solusi terbaik yang didapatkan saat ini	-
\vec{X}	vektor posisi yang akan diperbarui	-
\vec{D}'	jarak ikan paus ke- i terhadap target/mangsa (solusi terbaik)	-
b	konstanta untuk mendefinisikan bentuk spiral logaritma	-
l	bilangan <i>random</i> antara $[-1,1]$	-
p	bilangan <i>random</i> pada interval $0 \leq p \leq 1$ atau $[0,1]$	-
Y_{sp1}	<i>set point</i> komposisi distilat	mol
Y_{sp2}	<i>set point</i> komposisi bawah	mol
E_1	<i>error</i> komposisi distilat	mol
E_2	<i>error</i> komposisi bawah	mol
G_{c1}	pengendali pada komposisi distilat	-
G_{c2}	pengendali pada komposisi bawah	-
U_1	<i>output</i> pengendali pada komposisi distilat	mA
U_2	<i>output</i> pengendali pada komposisi bawah	mA
G_p	fungsi transfer	-

Y_1	komposisi distilat	mol
Y_2	komposisi bawah	mol

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Distilasi merupakan suatu proses pemisahan komponen yang didasarkan pada perbedaan volatilitas atau kecenderungan suatu komponen untuk menguap (Buckley et al., 1985). Proses distilasi sangat sulit untuk dikendalikan. Kesulitan dalam mengendalikan proses distilasi ini adalah dampak dari karakteristik yang dimiliki oleh proses distilasi itu sendiri. Proses distilasi memiliki karakteristik yang sangat tidak linier, berstruktur MIMO (*Multiple Input Multiple Output*), dan pada proses operasinya banyak terdapat gangguan. Kolom distilasi adalah contoh khas dari sistem MIMO dimana ada interaksi yang kuat antar variabel-variabelnya. Gangguan pada kolom distilasi biasanya datang dari berbagai sumber, seperti dari umpan (laju aliran umpan dan komposisi umpan), refluks, dan laju aliran uap. Kesulitan-kesulitan mengenai masalah sistem kendali pada kolom distilasi ini menarik banyak perhatian peneliti dari berbagai disiplin ilmu untuk berusaha memecahkannya (Mahmoud, 2018).

Kolom distilasi biasanya digunakan untuk proses purifikasi atau pemurnian suatu produk hasil reaksi antar reaktan, itu mengapa kolom distilasi sering kali ditemui di industri-industri kimia dan perminyakan. Tingkat kemurnian hasil produk kolom distilasi yang semakin tinggi akan menyebabkan kolom distilasi semakin sulit untuk dikendalikan. Hal ini disebabkan karena tingkat kemurnian produk sebanding dengan kondisi ketidaklinieran sistem dimana semakin tinggi tingkat kemurnian produk, maka semakin tidak linier kondisi yang terjadi pada sistem (Vasičkaninová et al., 2016). Sistem pengendalian pada kolom distilasi memiliki peran besar terhadap kualitas produk yang dihasilkan serta jumlah energi yang digunakan. Kolom distilasi memerlukan energi yang sangat besar pada proses pemanasan dan pendinginannya. Biaya yang diperlukan untuk memasok kebutuhan energi tersebut lebih dari 50% dari biaya total produksi, sehingga apabila kualitas hasil produk tidak sesuai dengan kriteria yang telah ditentukan, maka hasil produk akan diproses ulang dan menyebabkan biaya produksi semakin besar. Sistem

pengendalian yang baik pada sebuah kolom distilasi dapat menjaga dan meningkatkan kualitas hasil produksi, alhasil juga dapat menekan biaya produksi (Safitri et al., 2012).

Jenis pengendali yang sering diaplikasikan dalam sebuah sistem pengendalian adalah PID. Pengendali PID sudah banyak diterapkan pada berbagai jenis *plant* dan berbagai jenis permasalahan sistem pengendalian, termasuk diantaranya kolom distilasi. Performa dari pengendali PID pada sebuah sistem pengendalian, khususnya pada *plant* kolom distilasi dinilai kurang optimal. Hal ini disebabkan oleh banyaknya perubahan kritis pada proses kolom distilasi, sehingga pengendali PID tidak dapat mempertahankan komposisi setiap komponen hasil proses secara akurat (Guru et al., 2013).

Logika fuzzy merupakan salah satu jenis *artificial intelligence* yang dapat diaplikasikan untuk menyelesaikan berbagai masalah kehidupan, salah satunya dapat diaplikasikan sebagai pengendali. Logika fuzzy memiliki karakteristik yang mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks, sehingga cocok untuk diterapkan sebagai pengendali pada kolom distilasi yang bersifat sangat tidak linier (Safitri et al., 2012) (Kusumadewi & Purnomo, 2010). Kendala terbesar dalam mendesain sistem logika fuzzy sebagai pengendali adalah menentukan bentuk dan letak fungsi keanggotaan serta *rule base* yang optimal. Tidak ada aturan resmi dan pasti dalam penentuan bentuk fungsi keanggotaan, letak fungsi keanggotaan, dan *rule base* yang digunakan karena sistem logika fuzzy adalah sistem yang berbasis *human expert* (Safitri et al., 2012). Beberapa teknik optimasi seperti *genetic algorithm* (GA) (Safitri et al., 2012), *particle swarm optimization* (PSO) (Dorrah & El-garhy, 2012), dan berbagai jenis teknik optimasi lainnya digunakan untuk mengatasi kendala dalam menentukan parameter pengendali fuzzy yang optimal.

Whale Optimization Algorithm (WOA) adalah teknik meta-heuristik berbasis populasi yang saat ini diusulkan lebih unggul daripada PSO, *gravitational search algorithm* (GSA), *differential evolution* (DE), *fast evolutionary programming* (FEP), dan *evolution strategy with covariance matrix* (ESCMA) (Sahu et al., 2017). WOA merupakan salah satu algoritma optimasi yang dikembangkan oleh Mirjalili dan Lewis pada tahun 2016 dan didasari oleh tingkah laku paus dalam berburu.

WOA saat ini mulai diterapkan di berbagai masalah optimasi, salah satunya di bidang teknik/*engineering* (Sahu et al., 2017). Pada pengaplikasiannya, menurut (Gharehchopogh & Gholizadeh, 2019) WOA dapat memberikan hasil yang cukup bagus dan efisien di berbagai permasalahan optimasi.

Oleh karena itu, untuk mengurangi pemakaian energi, menekan biaya produksi, dan menghasilkan produk yang berkualitas tinggi dari sebuah kolom distilasi diperlukannya suatu sistem pengendalian yang tepat. Penelitian yang akan dilakukan pada tugas akhir ini adalah merancang sekaligus menganalisis performansi sistem pengendalian pada kolom distilasi menggunakan *fuzzy-whale optimization algorithm*. Karakteristik respons hasil pengendali dengan metode *fuzzy-whale optimization algorithm* akan dibandingkan dengan metode PID dan metode fuzzy yang tidak menggunakan algoritma optimasi untuk mengetahui kualitas performansi yang dihasilkan. Simulasi untuk mendapatkan grafik respons sistem pengendali dilakukan dengan menggunakan *software* Matlab.

1.2 Rumusan Masalah

Rumusan masalah dalam tugas akhir ini berdasarkan latar belakang yang telah terpapar di atas sebagai berikut:

1. Berapa nilai parameter pengendali *fuzzy-whale optimization algorithm* yang menghasilkan performansi terbaik?
2. Apakah desain pengendali berbasis *fuzzy-whale optimization algorithm* mampu meningkatkan performansi pada sebuah kolom distilasi?

1.3 Tujuan

Tujuan dalam tugas akhir ini berdasarkan permasalahan yang telah terpapar di atas sebagai berikut:

1. Mengetahui parameter pengendali berbasis *fuzzy-whale optimization algorithm* yang mampu menghasilkan performansi terbaik.
2. Mengetahui dan membandingkan hasil performansi pengendali *fuzzy-whale optimization algorithm* dengan hasil performansi pengendali fuzzy tanpa algoritma optimasi dan pengendali PID pada sebuah kolom distilasi.

1.4 Batasan Masalah

Batasan-batasan masalah dalam tugas akhir ini sebagai berikut:

1. Kolom distilasi yang digunakan sebagai objek penelitian pada tugas akhir ini adalah kolom distilasi biner methanol-air Wood & Berry.
2. Parameter pengendali yang dioptimasi dalam tugas akhir ini adalah titik batasan *error*, delta *error*, dan sinyal kontrol pada fungsi keanggotaan *input* dan *output*.
3. *Software* yang digunakan untuk mensimulasikan hasil perancangan sistem pengendalian komposisi pada kolom distilasi adalah *Software* Matlab.
4. Variabel yang dikendalikan adalah komposisi produk distilat dan komposisi produk bawah kolom distilasi dengan memanipulasi *reflux* dan *steam flow rate*.

1.5 Sistematika Laporan

Penelitian dalam tugas akhir ini terdiri dari lima BAB. BAB I berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika laporan penulisan tugas akhir ini. BAB II berisi tentang *review* jurnal atau tinjauan pustaka dan teori-teori yang berkaitan dengan tugas akhir ini. Alur beserta tahapan-tahapan yang dilakukan selama pengerjaan tugas akhir ini dibahas rinci pada BAB III. BAB IV berisi tentang hasil data yang didapatkan dari simulasi *software* Matlab untuk kemudian dilakukan analisis terhadap hasil yang didapatkan. BAB V berisi tentang kesimpulan dan juga saran dari hasil yang didapat pada tugas akhir ini.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai sistem pengendalian dengan menggunakan *fuzzy logic controller* (FLC) pada kolom distilasi telah dilakukan oleh (Vasičkaninová et al., 2016) dalam sebuah *paper*nya yang berjudul “*Fuzzy Control of a Distillation Column*”. (Vasičkaninová et al., 2016) menyajikan keefisienan fuzzy sebagai pengendali pada sebuah kolom distilasi dengan membandingkannya dengan hasil respons pengendali PID. Terdapat dua parameter yang dibandingkan mengenai performansi sistem pengendali yaitu dari segi temporal yang meliputi *settling time* dan dari segi spasial yang meliputi IAE, ISE, ITAE, dan *overshoot*. Bentuk fungsi keanggotaan pengendali fuzzy yang digunakan adalah bentuk segitiga, baik pada fungsi keanggotaan *input* maupun fungsi keanggotaan *output*. *Input* pengendali fuzzy yang digunakan sebanyak dua yaitu *error* dan *delta error*, dimana keduanya terdiri dari lima himpunan fuzzy, sedangkan *output* berjumlah satu dengan empat himpunan fuzzy. Respons dari pengendali fuzzy membuktikan keefektifannya dibandingkan dengan pengendali PID. Respons dari pengendali fuzzy mempunyai respons yang lebih cepat dibandingkan dengan respons PID dengan nilai *settling time* pengendali fuzzy dan PID berturut-turut sebesar 101 s dan 199 s. Pengendali fuzzy juga menghasilkan IAE, ISE, dan ITAE yang lebih kecil dibandingkan dengan pengendali PID. Berdasarkan respons pengendali yang didapat, disimpulkan bahwa sistem logika fuzzy lebih cocok dan lebih baik diimplementasikan sebagai pengendali pada masalah-masalah proses yang tidak linier dibandingkan dengan pengendali konvensional PID.

Penelitian lain mengenai sistem pengendalian pada kolom distilasi berbasis logika fuzzy juga dilakukan oleh (Safitri et al., 2012) dengan judul “Perancangan Sistem Kontrol *Genetic-Fuzzy*: Studi Kasus pada Pengendalian *Top and Side End Point Composition* dan *Bottom Refluks Temperature* pada *Distillation Column*”. Kendala dalam merancang sebuah sistem pengendali berbasis logika fuzzy adalah tidak adanya aturan khusus untuk menentukan parameter fuzzy yang optimal. Oleh

karena itu, (Safitri et al., 2012) menggunakan algoritma genetika/*genetic algorithm* (GA) untuk mengoptimasi letak dan rentang nilai dari fungsi keanggotaan fuzzy yang digunakan. Hasil yang didapat menunjukkan dan membuktikan bahwa pengendali fuzzy-GA dapat memperbaiki respons sistem. Waktu naik yang didapat dari setiap variabel kontrol adalah sebesar 64.7 s pada respons Y1, 212.8 s pada Y2, dan 11.83 s pada Y7.

Serupa dengan penelitian yang dilakukan oleh (Safitri et al., 2012), (Dorrah & El-garhy, 2012) menggunakan *Particle Swarm Algorithm* untuk mengoptimasi letak dan rentang nilai dari fungsi keanggotaan fuzzy yang digunakan. Dalam jurnalnya yang berjudul “*PSO based Optimized Fuzzy Controllers for Decoupled Highly Interacted Distillation Process*”, penggunaan FLC dilatarbelakangi oleh ketidakefisienan pengendali PID dalam mengendalikan suhu *tray* pada kolom distilasi *double*/ganda ketika sistem terganggu oleh faktor-faktor yang tidak diketahui. Hal tersebut disebabkan karena proses penyetelan parameter pengendali PID tergolong sulit dan membutuhkan waktu lama serta harus dilakukan secara benar dan sesuai dengan dinamika proses. Penelitian yang dilakukan oleh (Dorrah & El-garhy, 2012) mengembangkan penelitian sebelumnya dalam rangka mengoptimalkan desain pengendali dengan menggunakan *fuzzy PID controller*. Hasil yang didapat bahwa *PSO-based fuzzy PID controller* memiliki *sum square error* yang lebih kecil dibandingkan *PSO-based conventional PID controller*. Dari hasil yang didapat membuktikan *fuzzy PID controller* dengan optimasi algoritma PSO membuktikan kelayakan dan keunggulannya sebagai pengendali kolom distilasi *double*.

Penelitian yang menjadi referensi utama tugas akhir ini adalah penelitian yang dilakukan oleh (Qais et al., 2020) dan (Patel & Debnath, 2019). Kedua penelitian ini menyajikan bagaimana *whale optimization algorithm* (WOA) dapat mengoptimasi parameter sebuah pengendali logika fuzzy. (Qais et al., 2020) memanfaatkan WOA untuk mengoptimasi parameter pengendali fuzzy, yaitu berupa rentang dari fungsi keanggotaan *input error* dan *delta error*. Metode penalaran fuzzy yang digunakan ialah metode penalaran sugeno. Pengendali fuzzy ini diterapkan pada sebuah *plant*, yaitu *wind power plant* untuk meningkatkan kemampuan *fault ride* dari *wind power plant* yang terhubung ke sebuah sistem

jaringan listrik. Hasil simulasi menunjukkan bahwasanya pengendali fuzzy-WOA dapat menghasilkan waktu respons yang cepat, *error steady state* yang kecil, dan *maximum overshoot* yang rendah dibandingkan dengan fuzzy-GA dan fuzzy-GWO. Disisi lain, (Patel & Debnath, 2019) memanfaatkan WOA untuk mendapatkan nilai *gain factor* dari pengendali Fuzzy-PI yang diimplementasikan pada sistem *thermal-hydro-wind*. Hasil yang didapat menyatakan bahwa pengendali Fuzzy-PI berbasis WOA dapat menghasilkan respons yang baik. Hal ini dibuktikan dengan nilai *settling time* dan *maximum overshoot* yang didapat oleh Fuzzy-PI berbasis WOA lebih unggul daripada pengendali PID biasa.

Pengaplikasian WOA tidak hanya dapat digunakan untuk masalah optimasi sebuah sistem pengendali saja, melainkan juga dapat digunakan dan diaplikasikan pada berbagai jenis masalah kehidupan. WOA dapat diaplikasikan pada proses pengklasifikasian data, pengestimasi data, segmentasi gambar, dan lain sebagainya. Pada pengaplikasiannya, terkadang WOA sering kali digabungkan dengan sebuah sistem, seperti *artificial neural network*, *expert system*, dan berbagai jenis sistem lainnya. (Gharehchopogh & Gholizadeh, 2019) telah melakukan survei mengenai performa WOA saat diaplikasikan pada berbagai jenis masalah kehidupan. Jurnal tersebut menunjukkan bahwa WOA cukup efisien dalam menyelesaikan berbagai masalah kehidupan, khususnya pada masalah optimasi.

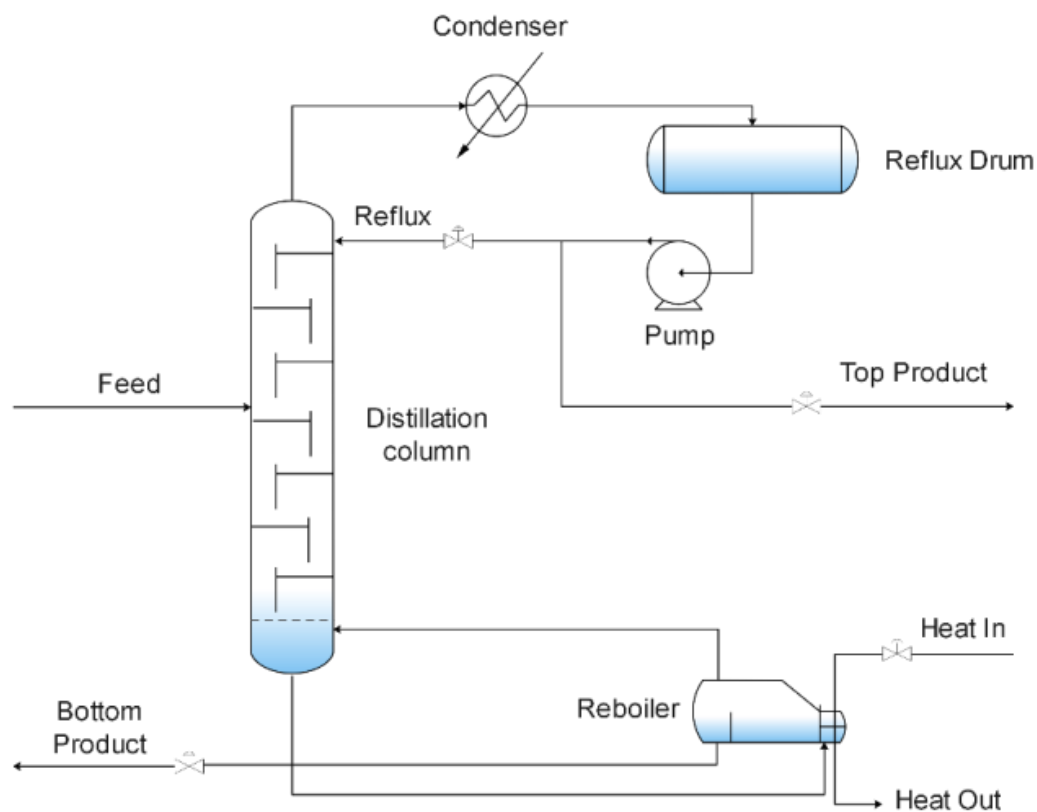
2.2 Kolom Distilasi Biner

Distilasi merupakan sebuah teknologi yang sudah sangat lama digunakan untuk proses pemisahan komponen yang didasarkan pada perbedaan volatilitas atau kecenderungan suatu komponen untuk menguap. Distilasi merupakan teknologi pemisahan yang paling penting di suatu industri saat ini. Distilasi sangat cocok digunakan untuk pemisahan dengan kemurnian tinggi karena setiap tingkat pemisahan dapat diperoleh dengan konsumsi energi yang tetap dengan meningkatkan jumlah tahap kesetimbangannya (Halvorsen & Skogestad, 2000).

Industri-industri kimia mengaplikasikan proses distilasi ini pada sebuah kolom yang disebut sebagai kolom distilasi. Kolom distilasi berfungsi untuk memisahkan komponen dari sebuah campuran. Pemisahan ini dilakukan berdasarkan perbedaan titik didid atau volatilitas suatu komponen pada campuran

tersebut. Komponen yang memiliki titik didih lebih tinggi akan tetap berfase cair sedangkan komponen yang memiliki titik didih lebih rendah akan menguap (Muzwar et al., 2014).

Salah satu jenis kolom distilasi adalah kolom distilasi biner. Kolom distilasi biner adalah sebuah kolom yang memiliki dua saluran produk, yaitu produk atas (distilat)/*top product* dan produk bawah/*bottom product* (Sutapa et al., 2017). Gambar proses kolom distilasi biner disertakan pada **Gambar 2.1** berikut:



Gambar 2.1 Kolom Distilasi Biner (Komariah et al., 2009)

2.2.1 Komponen Utama dan Operasi Dasar Kolom Distilasi

Sebuah kolom distilasi terdiri atas beberapa komponen utama. Berikut adalah komponen-komponen utama penyusun kolom distilasi beserta penjelasannya:

a. *Shell* vertikal

Shell vertikal adalah tempat terjadinya proses pemisahan komponen *liquid* (Komariah et al., 2009).

b. Internal Kolom

Internal kolom seperti *tray/plate* dan/atau *packing* berfungsi untuk meningkatkan derajat pemisahan komponen (Komariah et al., 2009).

c. *Reboiler*

Reboiler berfungsi untuk menyediakan penguapan yang cukup untuk proses distilasi (Komariah et al., 2009).

d. Kondenser

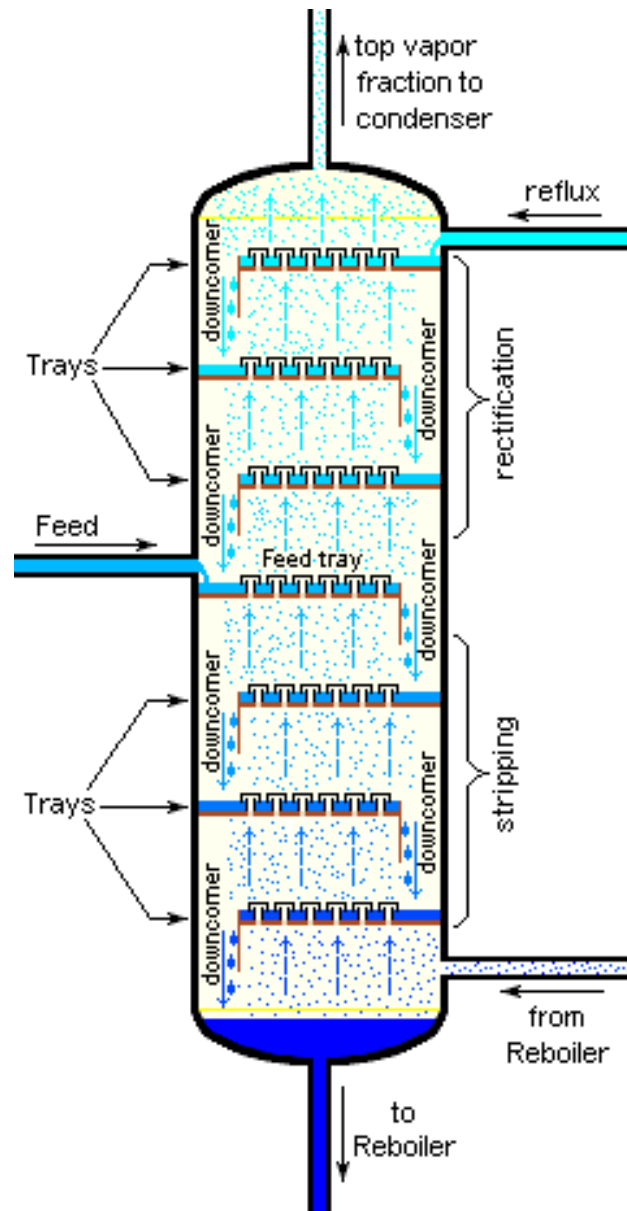
Kondenser berfungsi untuk mendinginkan dan mengkondensasikan uap yang keluar dari atas kolom (Komariah et al., 2009).

e. *Reflux Drum*

Reflux drum berfungsi untuk menampung uap yang terkondensasi dari kolom bagian atas sehingga cairan (*reflux*) dapat diumpankan kembali ke kolom (Komariah et al., 2009).

Operasi pada kolom distilasi diawali dengan masuknya komponen umpan/*feed* ke bagian tengah kolom pada bagian *tray* yang disebut dengan *feed tray*. *Tray* merupakan bagian dari kolom distilasi yang terletak di dalam kolom dan berfungsi sebagai daerah kontak fluida. *Feed* yang masuk kemudian mengalir menuju bawah kolom dan masuk ke dalam *reboiler* melalui bagian bawah *reboiler*. Proses pertukaran panas terjadi di dalam *reboiler* antara komponen umpan dengan *steam* pada *reboiler*, sehingga komponen umpan yang memiliki titik didik lebih rendah menguap dan masuk ke dalam kolom menuju bagian atas kolom. Uap tersebut kemudian didinginkan oleh kondenser sehingga menyebabkan terjadinya perubahan fase dari uap menjadi cair. Komponen tersebut kemudian ditampung dalam *reflux drum*, kemudian dikeluarkan dari *reflux drum* dan dikenal sebagai produk distilat. Sebagian komponen dalam *reflux drum* diumpankan kembali pada kolom distilasi guna meningkatkan efisiensi kolom distilasi dengan memberikan pendinginan kepada uap yang mengalir ke atas. Komponen yang memiliki titik didik lebih tinggi (tidak menguap saat terjadi proses pertukaran panas di dalam *reboiler*) akan dikeluarkan dari *reboiler*, komponen ini dikenal dengan sebutan produk bawah/*bottom product* (Komariah et al., 2009). Gambar mengenai ilustrasi

operasi atau proses yang terjadi pada sebuah kolom distilasi disajikan pada **Gambar 2.2** berikut :



Gambar 2.2 Ilustrasi Operasi Kolom Distilasi (Komariah et al., 2009)

2.2.2 Pemodelan Matematis

Model matematis kolom distilasi secara umum dituliskan dan diturunkan dengan menggunakan konsep hukum fisika. Konsep hukum fisika yang digunakan adalah hukum kesetimbangan massa, hukum kesetimbangan massa komponen, dan hukum kesetimbangan energi. Model fisis dari suatu proses harus dapat

menggambarkan karakteristik dinamis sistem tersebut secara memadai (Halvorsen & Skogestad, 2000). Model matematis diturunkan dari model fisis sistem. Model matematis kolom distilasi dengan menggunakan hukum kesetimbangan massa, hukum kesetimbangan massa komponen, dan hukum kesetimbangan energi pada keadaan *steady state* dituliskan pada persamaan (2.1) sampai dengan (2.3).

a. Kesetimbangan Massa Total Kolom Distilasi

$$F = D + B \quad \dots (2.1)$$

dimana : F = laju aliran *feed* (kg/s)

D = laju aliran *distillate product* (kg/s)

B = laju aliran *bottom product* (kg/s)

b. Kesetimbangan Massa Komponen Kolom Distilasi

$$F \cdot x_F = D \cdot x_D + B \cdot x_B \quad \dots (2.2)$$

dimana : x_F = fraksi *feed* (mol)

x_D = fraksi *distillate product* (mol)

x_B = fraksi *bottom product* (mol)

c. Kesetimbangan Energi Total Kolom Distilasi

$$F \cdot h_f - D \cdot h_D - B \cdot h_B + Q_r - Q_c = 0 \quad \dots (2.3)$$

dimana : h_f = entalpi pada *feed* (J/kg)

h_D = entalpi pada *distillate* (J/kg)

h_B = entalpi pada *bottom* (J/kg)

Q_r = *heatflow reboiler* (J/kg)

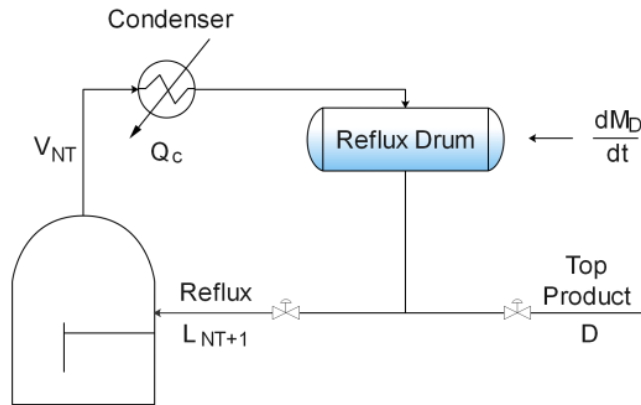
Q_c = *heatflow kondenser* (J/kg)

Pemodelan matematis juga dilakukan pada setiap komponen penyusun kolom distilasi. Pemodelan juga didapat dengan menggunakan konsep hukum kesetimbangan massa, hukum kesetimbangan massa komponen, dan hukum kesetimbangan energi (Muzwar et al., 2014). Berikut adalah pemodelan matematis dari komponen penyusun kolom distilasi, yaitu bagian atas kolom distilasi yang

disertai tangki akumulator dan kondenser dan juga bagian bawah kolom distilasi yang disertai *reboiler*:

1. Tangki Akumulator dan Kondenser

Mengetahui informasi mengenai variabel-variabel apa saja yang terlibat pada *plant* selama proses berlangsung merupakan hal yang penting dalam memodelkan sebuah *plant*. Gambar kolom distilasi pada *tray* bagian atas lengkap dengan informasi variabel *input-output* yang terlibat selama proses berlangsung disajikan pada **Gambar 2.3** berikut:



Gambar 2.3 Tangki Akumulator dan Kondenser (Biyanto et al., 2015)

Model matematis kolom distilasi pada *tray* paling atas, tangki akumulator, dan kondenser dituliskan pada persamaan (2.4) sampai dengan (2.6).

- a. Kestimbangan Massa pada Tangki Akumulator dan Kondenser

$$\frac{dM_D}{dt} = V_{NT} - L_{NT+1} - D \quad \dots (2.4)$$

dimana : M_D = total massa yang tertahan (kg)

V_{NT} = laju aliran massa uap (kg/s)

L_{NT+1} = laju aliran *reflux* (kg/s)

- b. Kestimbangan Massa Komponen pada Tangki Akumulator dan Kondenser

$$\frac{d(M_D \cdot x_D)}{dt} = V_{NT} \cdot y_{NT} - (L_{NT+1} + D) x_D \quad \dots (2.5)$$

dimana : y_{NT} = fraksi dari uap keluaran kolom distilasi (mol)

- c. Kestimbangan Energi pada Tangki Akumulator dan Kondenser

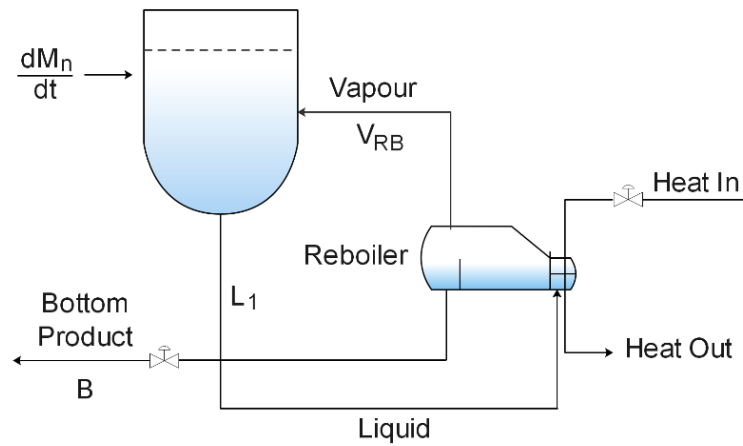
$$\frac{d(M_D \cdot h_D)}{dt} = V_{NT} \cdot H_{NT} - L_{NT+1} \cdot h_{NT+1} - D \cdot h_D + Q_C \quad \dots (2.6)$$

dimana : H_{NT} = entalpi pada uap keluaran *top stage* kolom (J/kg)

h_{NT+1} = entalpi pada *reflux* (J/kg)

2. Bagian Bawah Kolom

Sama halnya dengan memodelkan tangki akumulator dan kondenser, informasi mengenai variabel yang terlibat dalam proses penting untuk diketahui. Gambar kolom distilasi bagian bawah dengan informasi variabel *input-output* yang terlibat selama proses berlangsung disajikan pada **Gambar 2.4** berikut:



Gambar 2.4 Bagian Bawah Kolom Distilasi (Biyanto et al., 2015)

Pemodelan matematis kolom distilasi bagian bawah dituliskan pada persamaan (2.7) sampai dengan (2.9).

- a. Kestimbangan Massa pada Bagian Bawah Kolom Distilasi

$$\frac{dM_B}{dt} = (L_1 - V_{RB} - B) \quad \dots (2.7)$$

dimana : L_1 = laju aliran keluaran *bottom stage* kolom (kg/s)

V_{RB} = laju aliran uap dari *reboiler* (kg/s)

- b. Kestimbangan Massa Komponen pada Bagian Bawah Kolom Distilasi

$$\frac{d(M_B \cdot x_B)}{dt} = L_1 \cdot x_1 - V_{RB} y_B - B \cdot x_B \quad \dots (2.8)$$

dimana : x_1 = fraksi keluaran *bottom stage* kolom (mol)

y_B = fraksi uap dari *reboiler* (mol)

- c. Kestimbangan Energi pada Bagian Bawah Kolom Distilasi

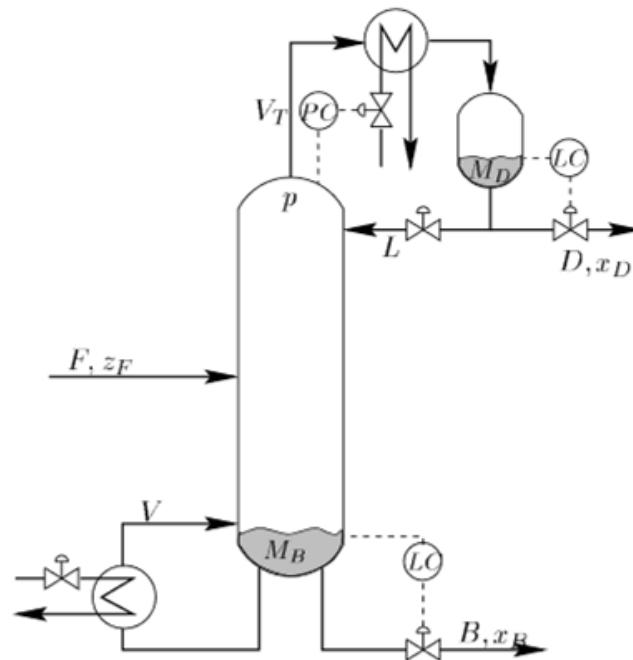
$$\frac{d(M_B \cdot h_B)}{dt} = L_1 \cdot h_1 - V_{RB} \cdot H_B - B \cdot h_B + Q_r \quad \dots (2.9)$$

dimana : h_1 = entalpi keluaran *bottom stage* kolom (J/kg)

H_B = entalpi uap dari *reboiler* (J/kg)

2.2.3 Struktur Kontrol L-V Kolom Distilasi Biner

Struktur kontrol L-V adalah struktur kontrol pada kolom distilasi yang menggunakan aliran *reflux* L untuk mengatur komposisi distilat x_D (*top product*) dan *heat input* Q_r yang masuk ke dalam *reboiler* untuk mengatur komposisi bawah x_B (*bottom product*) (Biyanto et al., 2015). Gambar struktur kontrol L-V kolom distilasi biner disajikan pada **Gambar 2.5** berikut:



Gambar 2.5 Struktur Kontrol L-V Kolom Distilasi (Skogestad, 1997)

2.3 Model *First Order Plus Death Time* FOPDT

Model *First Order Plus Death Time* (FOPDT) adalah model fungsi transfer orde satu yang disertai dengan waktu mati (*death time*) (Wara et al., 2016). Model FOPDT secara sederhana dapat merepresentasikan model dinamis suatu plant dengan cukup baik. FOPDT menyuguhkan cara sederhana untuk memisahkan dinamika proses menjadi *death time* dan fungsi alih orde satu. Oleh karenanya, banyak industri proses yang menggunakan model FOPDT dalam

merepresentasikan model dinamik *plant*nya (Roy & Iqbal, 2005). Model matematis proses FOPDT dapat direpresentasikan ke dalam persamaan *laplace* (2.10) berikut:

$$G(s) = \frac{y(s)}{u(s)} = \frac{K}{\tau s + 1} e^{-\theta s} \quad \dots (2.10)$$

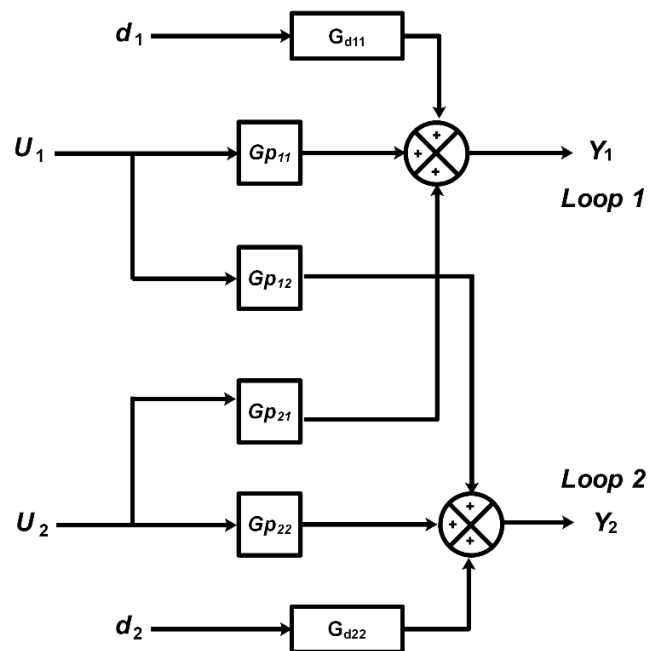
dimana : K = *gain* statis proses

τ = konstanta waktu tunak

θ = *death time*

2.4 Sistem Pengendalian *Multiple Input Multiple Output* 2x2 (MIMO)

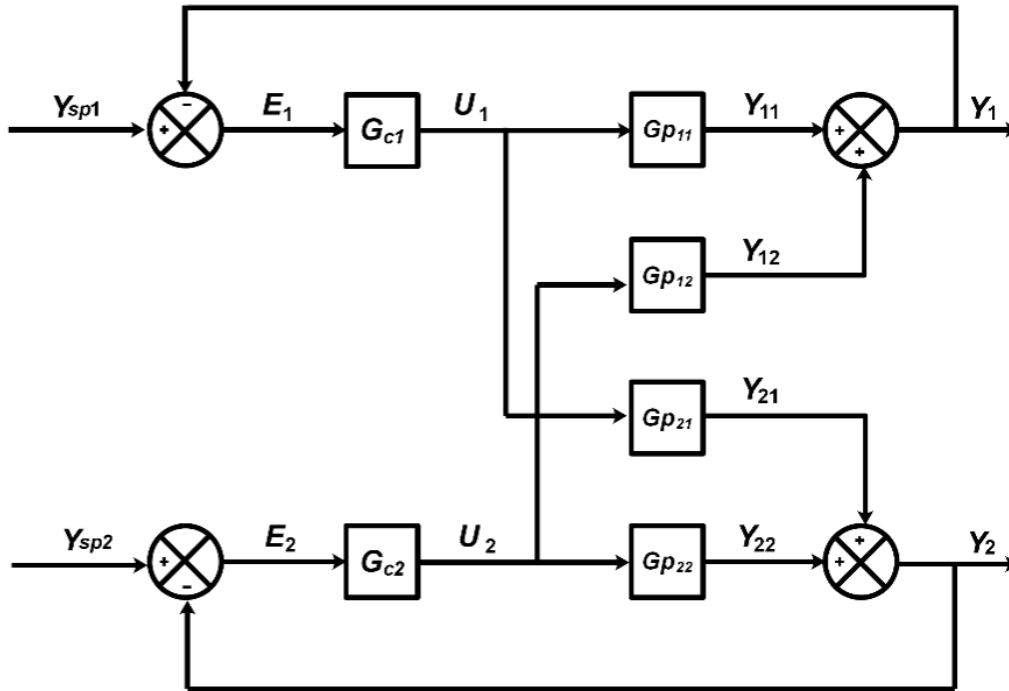
Sistem kendali MIMO adalah suatu sistem kendali dimana variabel *output* yang dikendalikan dan variabel *input* yang dimanipulasi berjumlah lebih dari satu. Salah satu ciri khas dari sistem kendali MIMO adalah adanya interaksi antara variabel yang dikendalikan dengan variabel yang dimanipulasi. Sistem MIMO 2x2 merupakan suatu sistem multivariabel yang memiliki dua variabel *output* yang dikendalikan (y) dan dua variabel *input* yang dimanipulasi (u) (Sutapa et al., 2017). Interaksi antara variabel yang dikendalikan dengan variabel yang dimanipulasi beserta *disturbance* pada sistem MIMO 2x2 ditunjukkan pada **Gambar 2.6** berikut:



Gambar 2.6 Interaksi Sistem MIMO 2x2 (Sutapa et al., 2017)

2.4.1 Sistem MIMO 2x2 Terdesentralisasi

Sistem kendali MIMO terdesentralisasi merupakan sistem kendali yang hanya menggunakan pasangan diagonal dari proses MIMO itu sendiri (Wara et al., 2016). Sistem ini terdiri dari pengendali umpan-balik independen yang mengendalikan *subset* dari variabel *output* yang dikendalikan dengan *subset* dari variabel *input* yang dimanipulasi. Sistem kendali MIMO terdesentralisasi tersusun oleh dua rangkaian sistem kendali SISO yang independen. Sistem kendali MIMO 2x2 terdesentralisasi memiliki dua struktur yang berbeda, yaitu struktur pengendali 1-1/2-2 dan struktur pengendali 1-2/2-1 (Sutapa et al., 2017). Sistem kendali MIMO terdesentralisasi dengan struktur 1-1/2-2 ditunjukkan pada **Gambar (2.7)** berikut:



Gambar 2.7 Diagram Blok Sistem Kendali MIMO 2x2 Terdesentralisasi
(Sutapa et al., 2017)

Struktur matriks dari fungsi alih proses pada sistem MIMO 2x2 dituliskan pada persamaan (2.11) berikut:

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \quad \dots (2.11)$$

2.4.2 Relative Gain Array (RGA)

Relative gain array (RGA) merupakan pendekatan yang paling sering digunakan untuk menentukan pasangan konfigurasi sistem kendali *multiloop* SISO pada sebuah sistem MIMO. RGA pada sistem FOPDT dapat dihitung dengan menggunakan *gain steady state* proses (Wara et al., 2016). *Gain steady state* proses pada sistem MIMO 2x2 dituliskan pada persamaan (2.12) berikut:

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \quad \dots (2.12)$$

Matriks RGA yang bersesuaian dengan *gain steady state* matriks dinyatakan pada persamaan (2.13) berikut:

$$RGA = \begin{bmatrix} \lambda_{11} & 1 - \lambda_{11} \\ 1 - \lambda_{11} & \lambda_{11} \end{bmatrix} \quad \dots (2.13)$$

dimana :

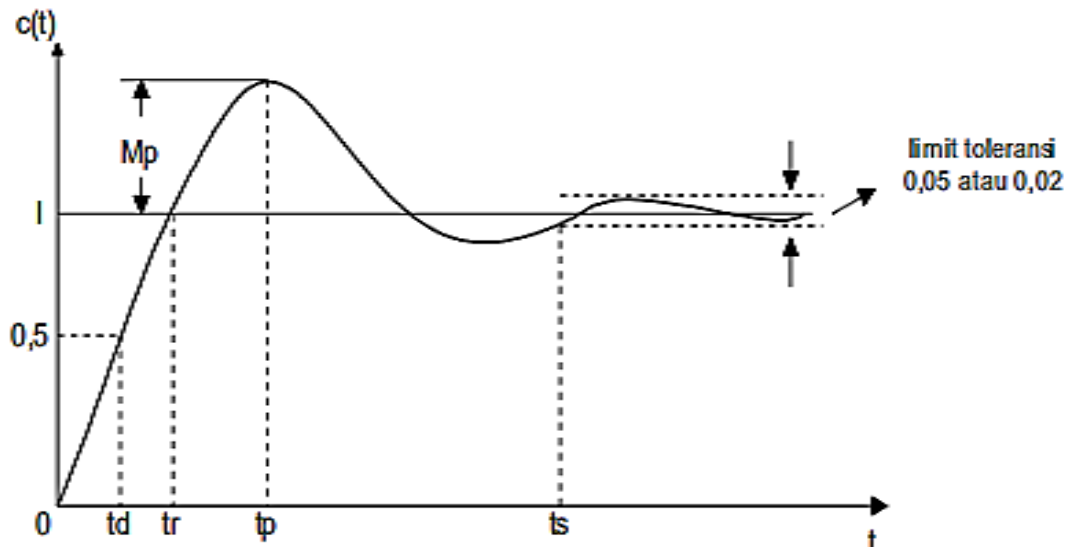
$$\lambda_{11} = \frac{1}{1 - \frac{K_{12} \cdot K_{21}}{K_{11} \cdot K_{22}}} \quad \dots (2.14)$$

λ bernilai 1 mengartikan bahwa pasangan variabel *input* dan *output* tidak memiliki interaksi. Nilai λ yang lebih dari 1 menandakan adanya interaksi pada sistem dan menyebabkan sistem tidak stabil, sedangkan λ bernilai negatif menandakan adanya nilai *inverse* (Chen & Seborg, 2002) (Wara et al., 2016).

2.5 Karakteristik Respons Sistem

Karakteristik suatu sistem pengendalian dicirikan oleh respons atau tanggapan transien terhadap *input*/masukan sinyal uji tangga satuan (*step*). Jika tanggapan terhadap sinyal *step* diketahui, maka secara matematis dapat dihitung tanggapan untuk sembarang masukan. Tanggapan transien suatu sistem pengendalian sering kali menunjukkan osilasi teredam sebelum mencapai keadaan mantapnya (Aisjah & Arifin, 2018). Suatu respons pada sebuah sistem pengendalian juga memuat informasi-informasi untuk mengetahui performansi atau kinerja yang dihasilkan oleh sistem pengendali, yaitu dari segi spasial maupun dari

segi temporal. Performansi dari segi spasial dapat dilihat dari nilai *maximum overshoot*nya, sedangkan performansi dari segi temporal dapat dilihat dari *time delay*, *rise time*, *peak time*, dan *settling time*.



Gambar 2.8 Karakteristik Respons Sistem Orde-2

(Husnaini & Krismadinata, 2017)

Gambar 2.8 adalah contoh gambar mengenai tanggapan transien suatu sistem terhadap sinyal *step*. Seperti yang telah disebutkan, bahwasanya sebuah respons suatu sistem pengendalian dapat menggambarkan dan memperlihatkan seberapa bagus performa atau kinerja dari suatu sistem pengendali. Dari respons yang dihasilkan oleh suatu sistem pengendalian dapat dihitung nilai performansinya. Berikut adalah penjelasan mengenai kriteria performansi yang terdapat dalam suatu kurva respons sistem orde-2 dari **Gambar 2.8** beserta model matematisnya:

1. *Time Delay*/Waktu Tunda (t_d)

Time delay adalah waktu yang diperlukan tanggapan untuk mencapai setengah nilai akhir (50% dari nilai akhir) untuk waktu yang pertama (Aisjah & Arifin, 2018). Persamaan matematis *time delay* ditunjukkan pada persamaan (2.15).

$$t_d = \frac{0.742}{\xi \omega_n} \quad \dots (2.15)$$

2. Rise Time/Waktu Naik (t_r)

Rise time adalah waktu yang dibutuhkan oleh tanggapan atau respons suatu sistem untuk naik dari 10% menuju 90% (Husnaini & Krismadinata, 2017). Rumus pendekatan atau model matematis untuk mendapatkan nilai *rise time* dinyatakan pada persamaan (2.16) berikut:

$$t_r = \frac{1}{\omega_d} \tan^{-1} \left(\frac{\omega_d}{-\sigma} \right) \quad \dots (2.16)$$

3. Maximum Overshoot (M_p)

Maximum Overshoot adalah nilai perbandingan antara nilai puncak tertinggi dari kurva tanggapan terhadap nilai akhir tertinggi (Husnaini & Krismadinata, 2017). Model matematis *maximum overshoot* ditunjukkan pada persamaan (2.17).

$$M_p = e^{-\xi \cdot \pi / \sqrt{1-\xi^2}} \times 100 \quad \dots (2.17)$$

Nilai puncak *overshoot* digunakan untuk menyatakan kestabilan relatif. Sistem dengan nilai *maximum overshoot* sekitar 40% mengindikasikan ketidakstabilan. Sistem biasanya memiliki nilai *maximum overshoot* 5-25%, terkadang juga sistem diharapkan tidak memiliki *overshoot* (Husnaini & Krismadinata, 2017).

4. Peak Time/Waktu Puncak (t_p)

Peak time adalah waktu yang dibutuhkan tanggapan untuk mencapai puncak *overshoot*. *Peak time* berkaitan dengan terjadinya puncak pertama kali (Aisjah & Arifin, 2018). Persamaan matematis dari *peak time* atau waktu puncak dituliskan pada persamaan (2.18) berikut:

$$t_p = \frac{\pi}{\omega_d} \quad \dots (2.18)$$

5. Settling Time/Waktu Turun (t_s)

Settling time merupakan waktu yang dibutuhkan agar kurva tanggapan mencapai dan tetap dalam daerah kriteria 2% atau 5% dari nilai akhir (Husnaini & Krismadinata, 2017). Rumus pendekatan atau model matematis untuk

mendapatkan nilai *settling time* dengan kriteria 2% dan *settling time* dengan kriteria 5% dinyatakan pada persamaan (2.19) dan (2.20) berikut:

- Kriteria 2%

$$t_s = \frac{4}{\xi \omega_n} \quad \dots (2.19)$$

- Kriteria 5%

$$t_s = \frac{3}{\xi \omega_n} \quad \dots (2.20)$$

dimana : ξ = rasio redaman

ω_n = frekuensi natural

ω_d = frekuensi redaman

2.6 Pengendali Konvensional *Proportional-Integral-Derivative* (PID)

Proportional-Integral-Derivative (PID) adalah sebuah metode pengendali yang digunakan pada sebuah sistem pengendalian *loop* tertutup (*close loop*). Metode ini terdiri dari tiga jenis pengendali, yaitu pengendali *proportional*, pengendali *integral*, dan pengendali *derivative*. Berikut adalah penjelasan dari masing masing pengendali:

1. Pengendali *Proportional*

Pengendali *proportional* merupakan pengendali sederhana yang sering digunakan pada berbagai macam permasalahan sistem kontrol. Pengendali *proportional* sangat bergantung pada besar nilai *error* sebuah sistem. Respons dari pengendali *proportional* dapat diatur dengan mengalikan nilai *error* sistem dengan sebuah konstanta K_p (*proportional gain*) (Tehrani & Mpanda, 2012). Persamaan pengendali *proportional* dituliskan pada persamaan (2.21) berikut:

$$P = K_p \cdot \text{error}(t) \quad \dots (2.21)$$

Sebuah konstanta K_p yang tinggi akan menghasilkan perubahan *output* yang besar dari setiap perubahan *error* yang terjadi, sebaliknya konstanta K_p yang kecil akan menghasilkan perubahan *output* yang kecil untuk nilai *error* yang besar.

Apabila nilai konstanta sangat tinggi maka akan menyebabkan sistem menjadi tidak stabil, sedangkan apabila nilai konstanta sangat kecil maka akan menghasilkan aksi kontrol yang kecil dalam merespons sistem yang memiliki *disturbance*. Oleh karena itu, pengendali *proportional* memiliki keunggulan untuk mempercepat waktu naik (*rise time*), namun tidak memperkecil nilai *error steady state* (Tehrani & Mpanda, 2012).

2. Pengendali *Integral*

Sebuah pengendali *integral* sebanding atau proporsional terhadap besarnya nilai *error* dan durasi nilai *error*. *Integral* pada sebuah pengendali PID adalah jumlah *error* sesaat dari waktu ke waktu dan memberikan akumulasi nilai *offset* yang seharusnya sudah diperbaiki sebelumnya. Keunggulan dari pengendali *integral* adalah mampu mengeliminasi nilai *error steady state*, akan tetapi hal tersebut dapat membuat respons transien menjadi buruk (Tehrani & Mpanda, 2012). Persamaan atau model matematis dari pengendali *integral* dituliskan pada persamaan (2.22) berikut:

$$I = K_i \int_0^t error(t) dt \quad \dots (2.22)$$

Keluaran dari pengendali *integral* membutuhkan selang waktu tertentu, sehingga pengendali *integral* cenderung memperlambat respons. Selain itu pengendali *integral* memiliki karakteristik dapat mempertahankan nilai sebelumnya apabila nilai *input error* berharga nol, tetapi nilai *error* tidak berharga nol, maka *output* pengendali akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya nilai *error* dan *integral gain*. Nilai konstanta K_i yang besar akan mempercepat hilangnya *offset*, akan tetapi semakin besar nilai konstanta K_i akan menyebabkan peningkatan osilasi dari sinyal *output* pengendali (Radita, 2017).

3. Pengendali *Derivative*

Parameter pengendali *derivative* pada sebuah *error* proses dapat dihitung dengan menentukan gradien dari nilai *error* dalam fungsi waktu dan mengalikannya dengan konstanta K_d (*derivative gain*). Keunggulan dari sebuah

pengendali *derivative* adalah meningkatkan stabilitas sistem, mengurangi nilai *maximum overshoot*, dan memperbaiki respons transien (Tehrani & Mpanda, 2012). Persamaan atau model matematis dari pengendali *derivative* dituliskan pada persamaan (2.23) berikut:

$$D = K_d \frac{derror(t)}{dt} \quad \dots (2.23)$$

Pengendali *derivative* tidak dapat menghasilkan *output* apabila tidak ada perubahan pada *inputnya* (nilai *error*). Apabila nilai *error* berubah terhadap waktu, maka *output* yang dihasilkan oleh pengendali bergantung pada nilai konstanta K_d dan laju perubahan nilai *error* (Radita, 2017).

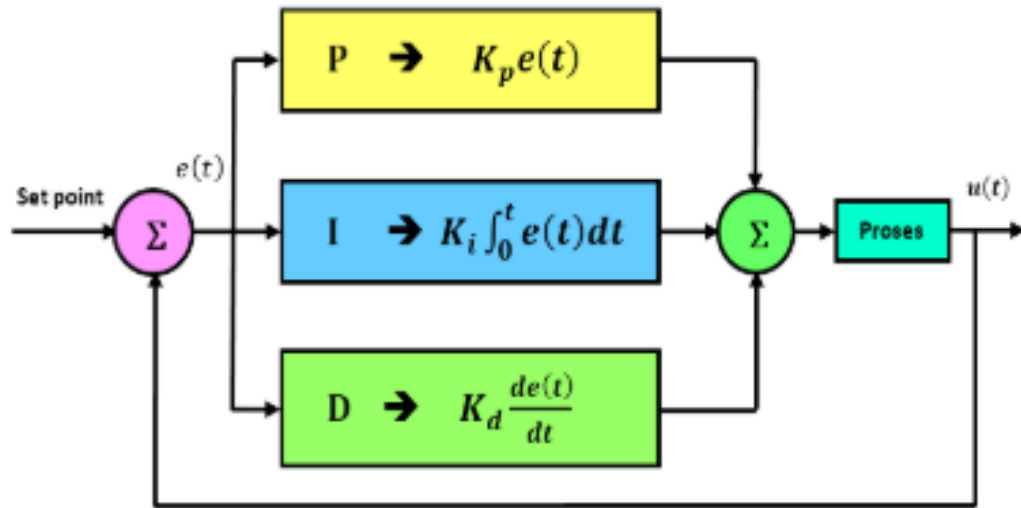
Masing-masing jenis atau tipe dari pengendali PID memiliki pengaruh yang berbeda-beda terhadap respons suatu sistem. Rangkuman mengenai pengaruh dari masing-masing jenis atau tipe pengendali PID terhadap karakteristik respons suatu sistem ditunjukkan pada **Tabel 2.1** berikut:

Tabel 2.1 Pengaruh PID terhadap Performansi Sistem (Tehrani & Mpanda, 2012)

Tipe Pengendali	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Steady State Error</i>
P	Decrease	Increase	Small Change	Decrease
I	Decrease	Increase	Increase	Decrease Significantly
D	Minor Decrease	Minor Increase	Minor Increase	No Effect in Theory

Suatu sistem pengendali PID memiliki *input* berupa sinyal *error* pada pengaplikasiannya. *Input* berupa sinyal *error* ini akan diproses oleh pengendali PID sesuai dengan tipe dari pengendali PID yang digunakan, tentunya dengan menggunakan persamaan matematis seperti yang telah ditunjukkan pada persamaan (2.21), persamaan (2.22) dan persamaan (2.23). Kemudian, pengendali PID akan

menghasilkan keluaran atau *output*. Keluaran atau *output* daripada pengendali PID ini akan memerintahkan aktuator untuk bertindak agar variabel *output* dari *plant* yang akan dikendalikan memiliki nilai *output* yang sesuai dengan *set point*. Untuk memperjelas proses sistem pengendalian berbasis PID, disajikanlah diagram blok sistem pengendaliannya.



Gambar 2.9 Blok Diagram Pengendali PID (Husnaini & Krismadinata, 2017)

Gambar 2.9 adalah gambar diagram blok secara umum suatu sistem pengendalian berbasis PID. Sistem pengendalian pada **Gambar 2.9** menggunakan pengendali tipe-PID. Dari **Gambar 2.9** didapatkan persamaan atau model matematis pengendali tipe-PID yang dituliskan pada persamaan (2.24) berikut:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad \dots (2.24)$$

dengan :

$$T_i = \frac{K_p}{K_i} \quad \dots (2.25)$$

$$T_d = \frac{K_d}{K_p} \quad \dots (2.26)$$

dan dari persamaan (2.24), didapatkan model fungsi transfer pengendali tipe-PID yang dituliskan pada persamaan (2.27) dan (2.28) berikut:

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad \dots (2.27)$$

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad \dots (2.28)$$

dimana : K_p = penguatan *proportional*

K_i = penguatan *integral*

K_d = penguatan *derivative*

T_i = waktu *integral*

T_d = waktu *derivative*

Ketiga parameter K_p , K_i , dan K_d biasanya disetel/dituning dalam rentang nilai yang diberikan, kegiatan tersebut biasa dikenal dengan *tuning* parameter pengendali. Dengan pilihan yang tepat dari kegiatan *tuning*, parameter pengendali dapat disesuaikan untuk *plant* tertentu untuk mendapatkan perilaku yang baik dari sistem yang dikendalikan (Tehrani & Mpanda, 2012). Beberapa metode *tuning* yang biasa digunakan adalah metode ziegler-nichols, cohen-coon, *autotune* pada *simulink* Matlab, dan lain sebagainya.

2.7 Sistem Logika Fuzzy

Sistem logika fuzzy atau yang biasa dikenal dengan *fuzzy logic* pertama kali diperkenalkan oleh Prof. Lutfi Asker Zadeh pada tahun 1965 di Universitas California, Berkeley. Lutfi Asker Zadeh memperkenalkan logika fuzzy melalui karya tulisannya mengenai teori himpunan fuzzy. Luthfi Asker Zadeh memodifikasi teori himpunan fuzzy dimana setiap anggota himpunan fuzzy memiliki derajat keanggotaan yang bernilai kontinyu antara 0 sampai dengan 1. Himpunan ini disebut dengan Himpunan Kabur (*Fuzzy Set*) (Setiawan et al., 2018).

Menurut (Kusumadewi & Purnomo, 2010), logika fuzzy didefinisikan sebagai suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. Ada beberapa alasan mengapa logika fuzzy banyak digunakan untuk menyelesaikan suatu masalah, antara lain:

1. Konsep logika fuzzy sederhana dan mudah dimengerti.
2. Logika fuzzy bersifat sangat fleksibel.

3. Logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat atau tidak akurat.
4. Logika fuzzy mampu memodelkan fungsi-fungsi *nonlinear* yang sangat kompleks.
5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan karena logika fuzzy berbasis *human expert*.
6. Logika fuzzy dapat digabungkan dengan teknik-teknik kendali konvensional.
7. Logika fuzzy didasarkan pada bahasa alami.

Sistem logika fuzzy memiliki beberapa istilah yang perlu diketahui untuk memahami lebih lanjut terkait sistem logika fuzzy itu sendiri. Istilah dalam sistem logika fuzzy diantaranya adalah variabel fuzzy, himpunan fuzzy, himpunan tegas (*crisp*), semesta pembicaraan, domain himpunan fuzzy, dan fungsi keanggotaan fuzzy. Berikut adalah penjelasan dari istilah-istilah yang terdapat dalam sistem logika fuzzy:

a. Variabel Fuzzy

Variabel fuzzy adalah variabel yang hendak dibahas dalam suatu sistem fuzzy (Kusumadewi & Purnomo, 2010). Lebih mudahnya variabel fuzzy merupakan *input* dan *output* yang akan dibahas pada sistem.

b. Himpunan Fuzzy

Himpunan fuzzy adalah suatu grup atau kelompok yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Himpunan fuzzy memiliki dua atribut, yaitu linguistik dan numeris. Linguistik merupakan penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, sedangkan numeris merupakan suatu nilai angka yang menunjukkan ukuran dari suatu variabel (Kusumadewi & Purnomo, 2010).

c. Himpunan Tegas (*Crisp*)

Himpunan tegas (*crisp*) merupakan nilai keanggotaan pada suatu item dalam suatu himpunan tertentu. Pada himpunan tegas, satu perubahan kecil pada item

tertentu dapat mengakibatkan perubahan item tersebut menjadi item lain (Yunita, 2016).

d. Semesta Pembicaraan

Semesta pembicaraan merupakan keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy (Wardani et al., 2017).

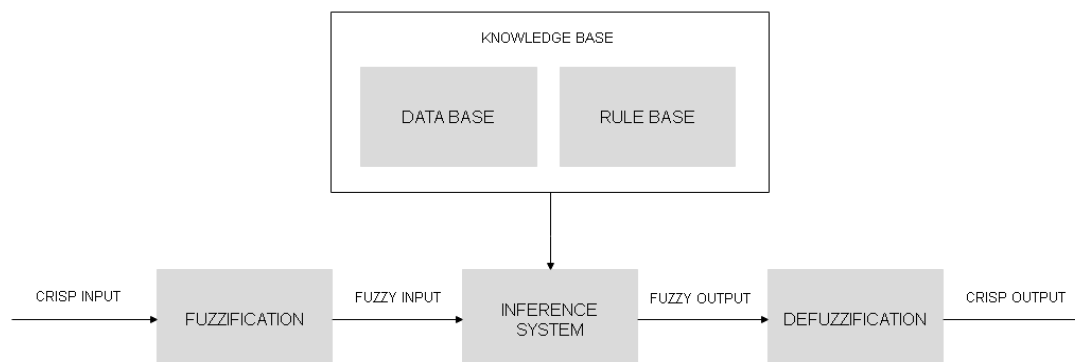
e. Domain Himpunan Fuzzy

Domain himpunan fuzzy merupakan keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy (Wardani et al., 2017).

f. Fungsi Keanggotaan

Fungsi keanggotaan atau *membership function* merupakan suatu kurva yang menunjukkan pemetaan titik-titik *input* ke dalam nilai keanggotaannya. Untuk mendapatkan nilai keanggotaan, salah satu caranya adalah dengan melalui pendekatan fungsi. Terdapat beberapa fungsi yang bisa digunakan, diantaranya adalah representasi kurva segitiga, representasi kurva trapesium, representasi kurva bahu, dan masih banyak yang lainnya (Wardani et al., 2017).

Secara umum, logika fuzzy mempunyai empat tahapan dalam menyelesaikan suatu masalah, yaitu fuzzifikasi, *knowledge base*, sistem inferensi, dan defuzzifikasi. Berikut adalah gambar diagram blok dari sistem logika fuzzy secara umum:



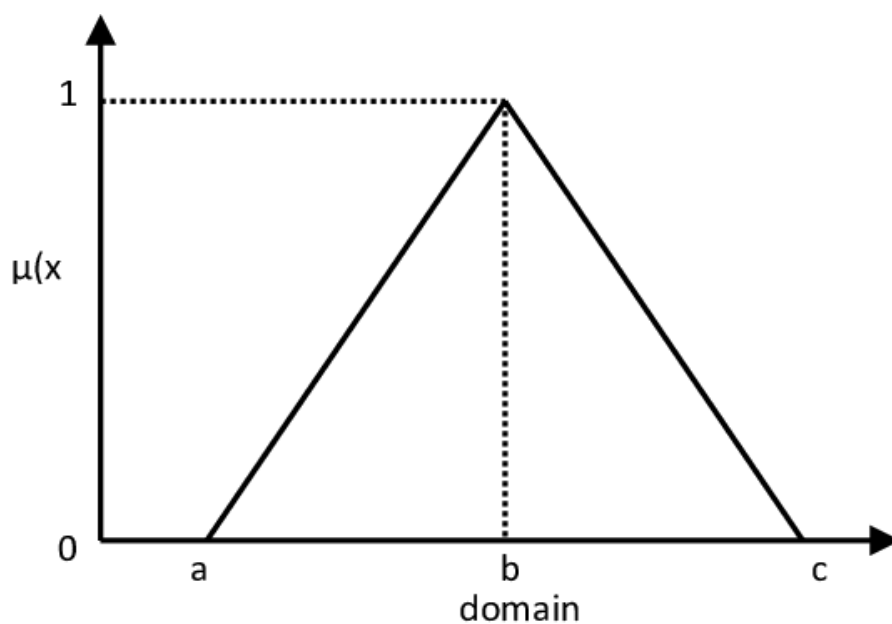
Gambar 2.10 Diagram Blok Sistem Logika Fuzzy

2.7.1 Fuzzifikasi

Fuzzifikasi adalah sebuah proses atau fase pertama pada perhitungan fuzzy untuk mengubah nilai *input* yang berupa bilangan *crisp* menjadi bilangan fuzzy (variabel linguistik). Proses fuzzifikasi ini dilakukan karena pada dasarnya sistem logika fuzzy adalah sistem yang berbasis variabel huruf atau variabel linguistik. Proses dari fuzzifikasi ini menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan fuzzy (Setiawan et al., 2018) (Taufiq, 2016). Fungsi keanggotaan dari suatu sistem logika fuzzy terdiri dari berbagai macam bentuk, beberapa bentuk fungsi keanggotaan dari sistem logika fuzzy adalah kurva segitiga, kurva trapesium, dan kurva bahu. Berikut adalah penjelasan secara singkat mengenai ketiga bentuk fungsi keanggotaan sistem logika fuzzy, yaitu fungsi keanggotaan fuzzy berbentuk segitiga, fungsi keanggotaan fuzzy berbentuk trapesium, dan fungsi keanggotaan fuzzy berbentuk bahu:

1. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan sebuah kurva yang terbentuk dari dua buah garis linier, yaitu garis linier naik dan garis linier turun (Taufiq, 2016). Representasi gambar dari fungsi keanggotaan berbentuk kurva segitiga ditunjukkan pada **Gambar 2.11** berikut:



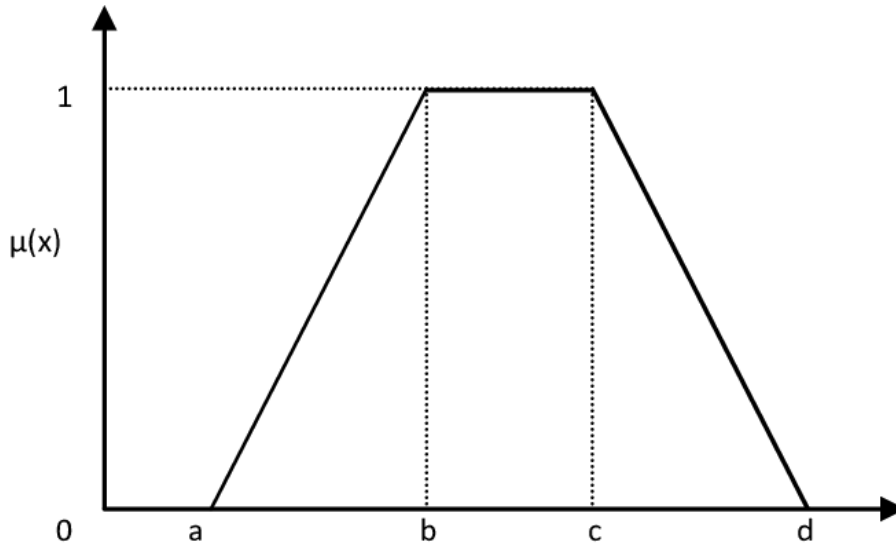
Gambar 2.11 Representasi Kurva Segitiga (Haris et al., 2010)

Fungsi keanggotaan pada representasi kurva segitiga digambarkan dalam persamaan (2.29) berikut:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ \frac{(b-x)}{(c-b)}; & b \leq x \leq c \end{cases} \quad \dots (2.29)$$

2. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya mirip dengan kurva segitiga, yaitu terdiri dari dua kurva linier, namun dipisahkan oleh satu buah garis lurus yang sejajar dengan sumbu x. Garis lurus tersebut mewakili nilai keanggotaan 1 (Haris et al., 2010). Representasi gambar dari fungsi keanggotaan berbentuk kurva trapesium ditunjukkan pada **Gambar 2.12** berikut:



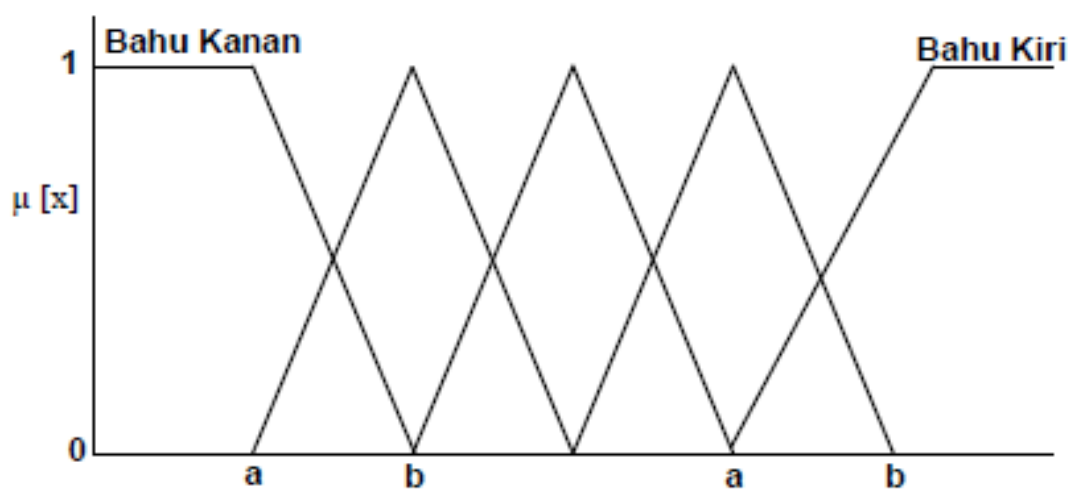
Gambar 2.12 Representasi Kurva Trapesium (Haris et al., 2010)

Fungsi keanggotaan pada representasi kurva trapesium digambarkan dalam persamaan (2.30) berikut:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{(d-x)}{(d-c)}; & c \leq x \leq d \end{cases} \quad \dots (2.30)$$

3. Representasi Kurva Bahu

Kurva bahu merupakan kurva gabungan dari kurva segitiga dan kurva trapesium. Pada kurva bahu, daerah yang terletak di tengah-tengah suatu variabel direpresentasikan oleh kurva berbentuk segitiga, pada sisi kanan dan kirinya akan naik turun, tetapi terkadang salah satu sisi dari variabel tidak mengalami perubahan. Himpunan fuzzy bahu, berbeda dengan himpunan fuzzy segitiga. Himpunan fuzzy bahu digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar (Praseptyo & Pujiyanta, 2014) (Hakim et al., 2015). Representasi gambar dari fungsi keanggotaan berbentuk kurva bahu ditunjukkan pada **Gambar 2.13** berikut:



Gambar 2.13 Representasi Kurva Bahu (Hasanah & Widiastuti, 2014)

2.7.2 Knowledge Base

Knowledge Base atau basis pengetahuan memuat teori-teori, fakta-fakta, pemikiran, dan hubungan antara satu dengan lainnya. Pengetahuan dari bidang tertentu disimpan dalam basis pengetahuan. Pengetahuan dibedakan menjadi dua, yaitu pengetahuan deklaratif dan prosedural. Pengetahuan deklaratif adalah fakta atau aturan khusus yang berhubungan dengan pokok permasalahan, sedangkan pengetahuan prosedural adalah pengetahuan tentang tindakan yang harus diambil untuk meraih sasaran tertentu. Cara lain untuk merepresentasikan pengetahuan

adalah dengan fungsi IF-THEN (Setiawan et al., 2018). Pada logika fuzzy terdapat tiga operasi dasar yaitu AND, OR, dan NOT (Agustin et al., 2016), berikut adalah penjelasannya:

1. Operator AND

Operator AND (*intersection*) berhubungan dengan operasi irisan (\cap) pada himpunan. α -predikat adalah hasil operasi operator AND yang didapat dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan (Wardani et al., 2017). Derajat keanggotaannya dirumuskan pada persamaan (2.31) berikut:

$$\mu_c(x) = \min(\mu_A(x), \mu_B(y)) \quad \dots (2.31)$$

2. Operator OR

Operator OR (*Union*) berhubungan dengan operasi gabungan pada himpunan. α -predikat adalah hasil operasi operator OR yang didapat dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan (Wardani et al., 2017). Derajat keanggotaannya dirumuskan pada persamaan (2.32) berikut:

$$\mu_c(x) = \max(\mu_A(x), \mu_B(y)) \quad \dots (2.32)$$

3. Operator NOT

Operator NOT berhubungan dengan operasi komplemen pada himpunan. α -predikat adalah hasil operasi operator NOT yang didapat dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1 (Wardani et al., 2017). Derajat keanggotaannya dirumuskan pada persamaan (2.33) berikut:

$$\mu_{A'}(x) = 1 - \mu_A(x) \quad \dots (2.33)$$

2.7.3 Sistem Inferensi

Sistem Inferensi Fuzzy atau *Fuzzy Inference System* adalah sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia dalam melakukan penalaran dengan nalurinya. Dengan menggunakan aturan pada *knowledge base* dan *database* sistem menduga fakta yang harus diperoleh atau dikumpulkan dengan kata lain mesin inferensi ini bertugas untuk membuat kesimpulan dari fakta yang diketahui sebelumnya (Setiawan et al., 2018) (Harefa, 2017). Metode mamdani dan sugeno adalah contoh dari sistem inferensi fuzzy. Berikut adalah penjelasan lebih rinci mengenai sistem inferensi fuzzy metode mamdani dan sistem inferensi fuzzy metode sugeno:

1. Metode Mamdani

Metode Mamdani atau yang biasa dikenal dengan nama Metode Max-Min diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 (Kusumadewi & Purnomo, 2010). Pada metode ini, diperlukan 4 tahapan untuk memperoleh *output* yaitu:

- a. Pembentukan himpunan fuzzy pada metode mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan fuzzy.
- b. Aplikasi fungsi implikasi (aturan) pada metode mamdani, fungsi implikasi yang digunakan adalah MIN.
- c. Komponen aturan Pada tahapan ini sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem fuzzy, yaitu: max, *additive*, dan probabilistik OR. Pada metode max, solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy dan mengaplikasikannya ke *output* dengan menggunakan operator OR (*union*).
- d. Penegasan atau defuzzifikasi *Input* dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain

himpunan fuzzy tersebut. Jika diberikan suatu himpunan fuzzy dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*.

2. Metode Sugeno

Penalaran sugeno diperkenalkan pertama kali oleh Takagi-Sugeno pada tahun 1985. Metode sugeno ini hampir sama dengan Metode Mamdani, namun metode sugeno memiliki perbedaan dengan metode mamdani. Hal yang membedakan metode sugeno dengan metode mamdani adalah *output* dari sistem dengan metode sugeno tidak berupa himpunan fuzzy, melainkan berupa persamaan linier atau konstanta (*singletone*) (Kusumadewi & Purnomo, 2010). Suatu ciri khas aturan fuzzy dalam model fuzzy Sugeno dibentuk dengan :

$$\text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = f(x,y)$$

dimana A dan B himpunan fuzzy dalam anteseden dan z merupakan $f(x,y)$ fungsi tegas dalam konsekuen. Jika $f(x,y)$ polinomial orde satu, FIS (*Fuzzy Inference System*) yang dihasilkan disebut model fuzzy Sugeno orde satu. Jika f konstan, dihasilkan model fuzzy Sugeno orde nol (Sitio, 2018).

2.7.4 Defuzzifikasi

Defuzzifikasi adalah kebalikan dari fuzzifikasi yaitu merupakan proses perubahan bilangan fuzzy keluaran sistem inferensi menjadi bilangan *crisp*. Defuzzifikasi juga sama dengan fuzzifikasi yaitu menggunakan fungsi keanggotaan dalam prosesnya (Setiawan et al., 2018). Terdapat beberapa metode untuk mendapatkan nilai *crisp* pada proses defuzzifikasi, baik pada penalaran mamdani maupun pada penalaran sugeno.

Penalaran mamdani memiliki beberapa metode defuzzifikasi, diantaranya yaitu metode *centroid of area* (COA), bisektor, *mean of maximum* (MOM), *largest of maximum* (LOM), dan *smallest of maximum* (SOM) (Sutikno & Waspada, 2012) sedangkan pada penalaran sugeno proses defuzzifikasi dilakukan dengan metode *Weighted Average* (WA) (Dorteus, 2015). Berikut adalah penjabaran atau

penjelasan dari masing-masing metode defuzzifikasi pada penalaran mamdani dan sugeno:

a. Metode *Centroid of Area* (COA)

Metode *centroid of area* (COA) adalah solusi untuk mendapatkan nilai *crisp* dengan cara mengambil titik pusat daerah fuzzy (Sutikno & Waspada, 2012). Model matematis dari metode COA dituliskan pada persamaan (2.34) dan (2.35).

- Untuk variabel kontinyu

$$z = \frac{\int_z z\mu(z)dz}{\int_z \mu(z)dz} \quad \dots (2.34)$$

- Untuk variabel diskrit

$$z = \frac{\sum_{j=1}^n z_j \mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \quad \dots (2.35)$$

b. Metode Bisektor

Metode bisektor adalah solusi untuk mendapatkan nilai *crisp* dengan mengambil nilai pada *domain* fuzzy yang memiliki nilai keanggotaan separuh dari jumlah total nilai keanggotaan pada daerah fuzzy (Sutikno & Waspada, 2012). Model matematis metode bisektor dituliskan pada persamaan (2.36) berikut:

$$\int_{\alpha}^{z^{BOA}} \mu(z)dz = \int_{z^{BOA}}^{\beta} \mu(z)dz \quad \dots (2.36)$$

dimana : $\alpha = \min\{z|z \in Z\}$

$\beta = \max\{z|z \in Z\}$

c. Metode *Mean of Maximum* (MOM)

Metode MOM ini adalah salah satu cara untuk mendapatkan nilai *crisp* kembali dengan mengambil nilai rata-rata *domain* yang memiliki nilai keanggotaan maksimum (Sutikno & Waspada, 2012).

d. Metode *Largest of Maximum* (LOM)

Metode LOM adalah solusi untuk mendapatkan nilai *crisp* dengan cara mengambil nilai terbesar dari *domain* yang memiliki nilai keanggotaan maksimum (Sutikno & Waspada, 2012).

e. Metode *Smallest of Maximum* (SOM)

Metode SOM adalah kebalikan dari metode LOM. Apabila pada metode LOM mengambil nilai terbesar dari *domain* untuk mendapatkan nilai *crisp* kembali, metode ini adalah kebalikannya yaitu mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum (Sutikno & Waspada, 2012).

f. *Weighted Average* (WA)

Weighted average (WA) atau rata-rata tertimbang/terbobot adalah metode defuzzifikasi yang dilakukan dengan cara menghitung nilai rata-rata dengan memperhitungkan bobot setiap data (Dorteus, 2015). Model matematis dari WA dituliskan pada persamaan (2.37) berikut:

$$WA = \frac{a_1z_1 + a_2z_2 + a_3z_3 + \dots + a_nz_n}{a_1 + a_2 + a_3 + \dots + a_n} \quad \dots (2.37)$$

dimana : WA = nilai rata-rata

a_n = nilai predikat aturan ke-n

z_n = indeks nilai *output* (konstanta) ke-n

2.8 Mean Squared Error (MSE)

Mean Squared Error merupakan metode untuk mengevaluasi metode peramalan, dimana masing-masing nilai kesalahan atau *error* yang didapat dari sebuah metode peramalan dikuadratkan. Pendekatan ini mengatur kesalahan peramalan yang besar dikarenakan nilai *error* yang didapat dari sebuah metode peramalan dikuadratkan (memberikan bobot yang besar). Konsekuensi dari metode ini adalah nilai *error* yang kecil akan semakin kecil dan nilai *error* yang besar akan semakin besar. MSE sangat baik dalam memberikan gambaran terhadap seberapa konsisten model yang dibangun (Margi S & Pendawa W, 2015) (Qodari, 2015). Rumus dari *mean squared error* dituliskan pada persamaan (2.38) berikut:

$$MSE = \sum_{i=1}^n \frac{(x_i - F_i)^2}{n} \quad \dots (2.38)$$

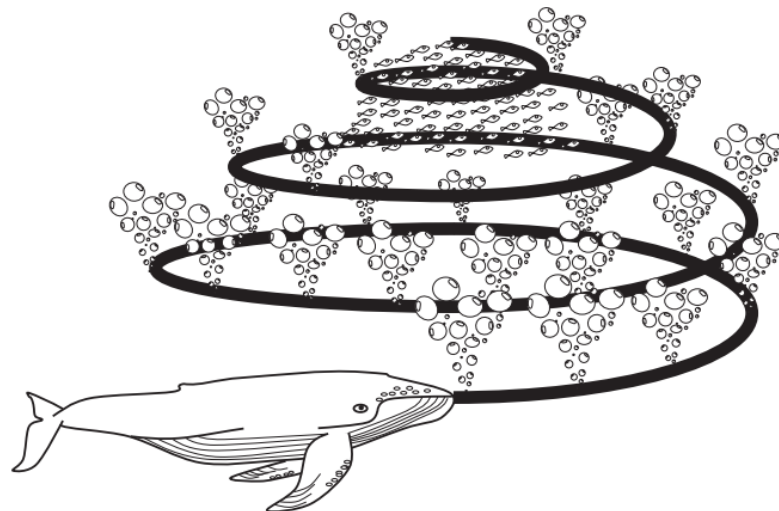
dimana : x_i = nilai data asli amatan ke-i

F_i = nilai ramalan amatan ke-i

n = banyaknya data

2.9 Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm (WOA) adalah sebuah algoritma optimasi berjenis *meta-heuristic* yang terinspirasi dari perilaku paus *humpback*/bungkuk dalam berburu (Sivalingam et al., 2017) (Gharehchopogh & Gholizadeh, 2019). Paus bungkuk adalah salah satu paus terbesar yang memiliki mangsa favorit berupa kawanan ikan kecil (Sahu et al., 2017). Dalam berburu, paus bungkuk memiliki teknik yang khas, teknik tersebut dikenal dengan sebutan teknik *bubble-net*. Sekumpulan paus bungkuk akan mengelilingi target atau mangsa sambil mengeluarkan gelembung dari mulutnya. Jarak diameter paus dengan mangsa semakin lama akan semakin kecil untuk mempersempit ruang gerak mangsa dan gelembung yang dikeluarkan akan menjadi jaring untuk menangkap mangsa. Kemudian, salah satu paus bungkuk akan berenang ke atas melewati jaring gelembung dan menangkap semua mangsa yang terjaring sebelumnya (Gharehchopogh & Gholizadeh, 2019). Teknik berburu paus bungkuk itulah yang menjadi inspirasi algoritma ini.



Gambar 2.14 Teknik *Bubble-net* Paus Bungkuk

(Gharehchopogh & Gholizadeh, 2019)

WOA memiliki tiga tahapan yang bersesuaian dengan tahapan paus bungkuk dalam menangkap mangsa, yaitu *encircling prey*/mengelilingi mangsa, *bubble-net*

attacking (fase eksploitasi), dan *search for prey*/pencarian mangsa (fase eksplorasi). Berikut adalah penjelasan secara rinci dari tahapan-tahapan algoritma WOA beserta model matematisnya:

1. *Encircling Prey*/Mengelilingi Mangsa

Tahap *encircling prey* atau tahap mengelilingi mangsa adalah tahap dimana paus bungkuk akan mengenali lokasi dari target/mangsa, kemudian paus bungkuk akan mengelilingi target/mangsa tersebut. Karena posisi desain optimal di dalam ruang pencarian tidak mungkin untuk diketahui secara priori, maka WOA mengasumsikan target/mangsa adalah kandidat solusi yang terbaik atau mendekati optimum. Pada tahap ini, dilakukan penentuan dan penguraian lebih lanjut mengenai *search agent* terbaik. Posisi dari *search agent* terbaik akan menjadi patokan untuk *search agent* yang lain dalam memperbarui posisinya menuju posisi optimum pada fase berikutnya (Gharehchopogh & Gholizadeh, 2019). Perilaku *search agent* dalam memperbarui posisi dimodelkan dalam persamaan (2.40) berikut:

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad \dots (2.39)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad \dots (2.40)$$

dimana : t = iterasi saat ini

\vec{A} = vektor koefisien

\vec{C} = vektor koefisien

\vec{X}^* = vektor posisi dari solusi terbaik yang didapatkan saat ini

\vec{X} = vektor posisi yang akan diperbarui

\vec{X}^* harus diperbarui pada setiap iterasi apabila solusi yang lebih baik dari iterasi sebelumnya ditemukan. Vektor koefisien \vec{A} dan \vec{C} dirumuskan pada persamaan (2.41) dan (2.42) berikut:

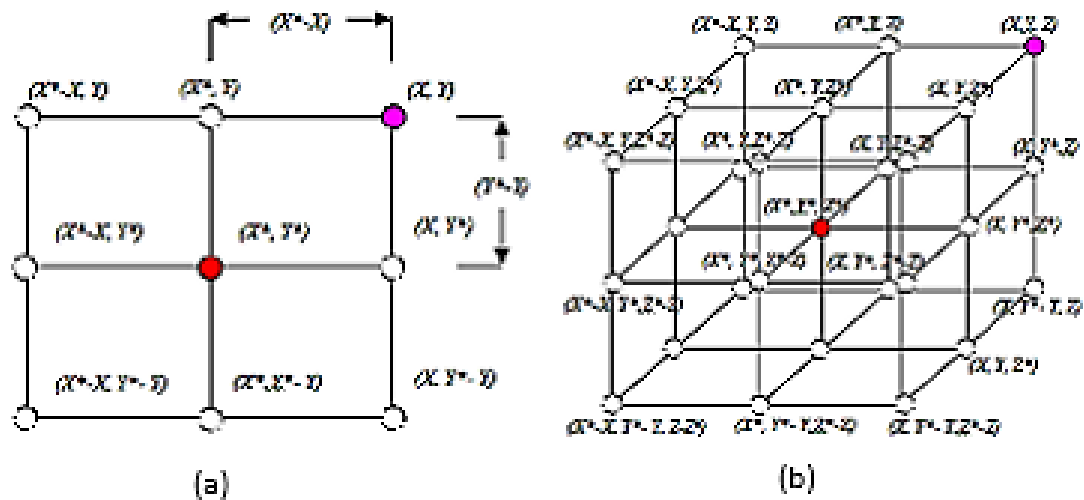
$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad \dots (2.41)$$

$$\vec{C} = 2\vec{r} \quad \dots (2.42)$$

dimana : \vec{a} = pengurangan linier dari 2 ke 0 selama iterasi (pada kedua fase)

\vec{r} = vektor *random* dalam [0,1]

Seperti yang telah disebutkan, bahwasanya persamaan (2.40) merupakan bentuk matematis dari perilaku *search agent* dalam memperbarui posisi. Untuk memperjelas persamaan (2.40), dibuatlah gambar untuk mengilustrasikan persamaan (2.40). Persamaan (2.40) pada permasalahan dua dimensi dan tiga dimensi diilustrasikan pada **Gambar 2.15**.



Gambar 2.15 (a) Posisi Vektor 2D (b) Posisi Vektor 3D dan Kemungkinan Posisi yang Akan Datang (Gharehchopogh & Gholizadeh, 2019)

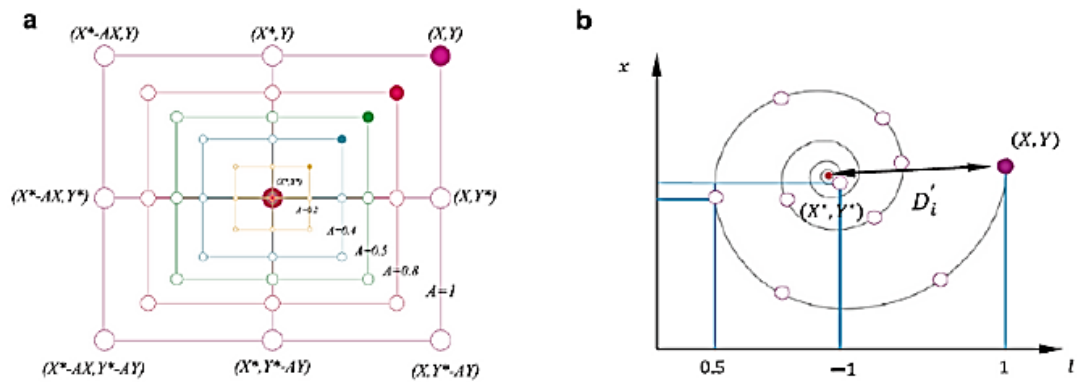
Pada permasalahan dua dimensi dalam **Gambar 2.15** (a), posisi (X, Y) dari sebuah *search agent* dapat diperbarui berdasarkan posisi terbaik saat ini (X^*, Y^*) . Berbagai posisi yang berada di sekitar *agent* terbaik didapat dengan memperhatikan posisi *agent* lain saat ini dan mengatur nilai vektor \vec{A} dan \vec{C} . **Gambar 2.15** (b) mengilustrasikan kemungkinan pergerakan posisi dari *search agent* pada permasalahan tiga dimensi. *Search agent* dapat mencapai berbagai posisi pada ruang lingkup pencarian dengan mendefinisikan *random* vektor (\vec{r}) (Gharehchopogh & Gholizadeh, 2019).

Maka dari itu dapat disimpulkan bahwa persamaan (2.40) memungkinkan beberapa paus atau *search agent* lain untuk memperbarui posisinya agar dapat mendekati posisi *search agent* terbaik saat ini. Sehingga, pergerakan dari kumpulan

search agent tersebut akan mengelilingi target/mangsa. Konsep yang sama juga dapat diterapkan pada sebuah ruang pencarian berdimensi- n dan *search agent* dapat bergerak di sekitar solusi terbaik yang didapat saat ini dalam sebuah *hyper-cubes* (Gharehchopogh & Gholizadeh, 2019).

2. Bubble-net Attacking (Fase Eksploitasi)

Tahap ini merupakan fase eksploitasi yang memiliki dua pendekatan untuk merumuskan model matematika dari perilaku paus dalam menyerang target/mangsa dengan teknik *bubble-net*. Dua pendekatan tersebut adalah *shrinking encircling mechanism* dan *spiral updating position* (Gharehchopogh & Gholizadeh, 2019). Penjelasan rinci beserta ilustrasi yang ditunjukkan pada **Gambar 2.16** mengenai kedua pendekatan, yaitu *shrinking encircling mechanism* dan *spiral updating position* adalah sebagai berikut:



Gambar 2.16 (a) *Shrinking Encircling Mechanism* (b) *Spiral Updating Position*

(Morshidi et al., 2018)

a. *Shrinking Encircling Mechanism*

Model perilaku ini didapatkan dengan mengurangi nilai dari \vec{a} pada persamaan (2.41). Rentang nilai dari fluktuasi \vec{A} juga akan berkurang seiring dengan berkurangnya nilai dari \vec{a} . Dengan kata lain, \vec{A} adalah nilai *random* pada interval $[-a, a]$, dimana a berkurang secara linier dari 2 ke 0 pada setiap iterasi. Posisi terbaru dari *search agent* dapat ditentukan di suatu tempat antara posisi asli *agent* dan posisi *agent* terbaik saat ini dengan mengatur nilai *random* \vec{A} pada $[-1, 1]$. Pada

Gambar 2.16 (a) terlihat bahwa pergerakan posisi (X, Y) terhadap (X^*, Y^*) dapat dicapai pada *range* $0 \leq A \leq 1$ dalam sebuah ruang dua dimensi (Gharehchopogh & Gholizadeh, 2019).

b. *Spiral Updating Position*

Pendekatan pada *spiral updating position* dilakukan dengan menghitung posisi *search agent* (X, Y) ke posisi dimana terdapat target/mangsa (X^*, Y^*) . Persamaan spiral yang dibuat antara posisi ikan paus (*search agent*) dengan target seperti perilaku pergerakan spiral paus bungkuk ditunjukkan pada persamaan (2.43).

$$\vec{X}(t+1) = \vec{D}^i \cdot e^{bl} \cdot \cos(2\pi d) + \vec{X}^*(t) \quad \dots (2.43)$$

$$\vec{D}^i = |\vec{X}^*(t) - \vec{X}(t)| \quad \dots (2.44)$$

dimana : \vec{D}^i = jarak ikan paus ke- i terhadap target/mangsa (solusi terbaik)

b = konstanta untuk mendefinisikan bentuk spiral logaritma

l = bilangan *random* antara $[-1, 1]$

Secara bersamaan, ikan paus bungkuk berenang mengelilingi mangsanya dengan melintasi jalur spiral dan mempersempit ruang gerak mangsanya (memperkecil diameter jarak ikan paus dengan mangsa). Untuk memodelkan perilaku simultan ini, diasumsikan terdapat peluang sebesar 0,5 ikan paus ke- i memilih antara mekanisme *shrinking encircling* atau *spiral updating position* untuk memperbarui posisinya selama iterasi (Gharehchopogh & Gholizadeh, 2019). Model matematisnya ditunjukkan pada persamaan (2.45) berikut:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0,5 \\ \vec{D}^i \cdot e^{bl} \cdot \cos(2\pi d) + \vec{X}^*(t) & \text{if } p \geq 0,5 \end{cases} \quad \dots (2.45)$$

dimana : p = bilangan *random* pada interval $0 \leq p \leq 1$ atau $[0, 1]$

3. Search for Prey (Fase Eksplorasi)

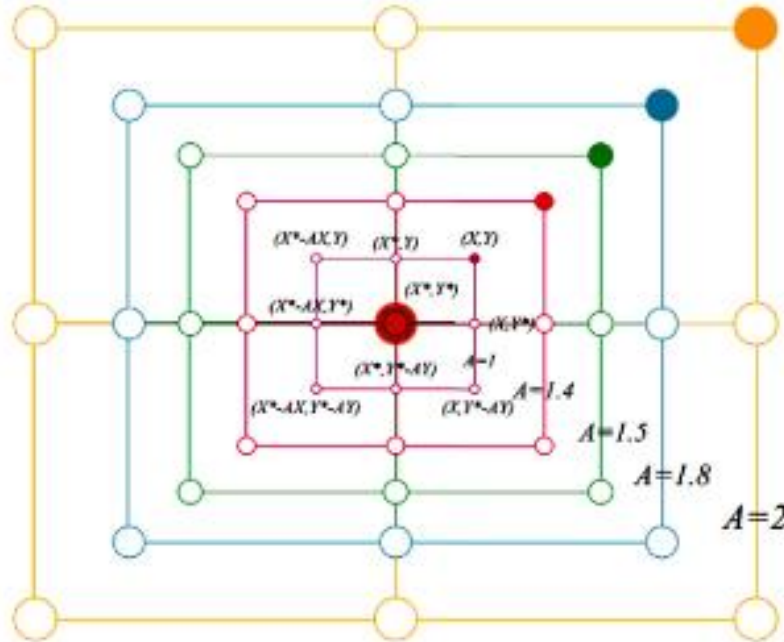
Sebuah pendekatan berbasis vektor variasi \vec{A} dapat digunakan dan dimanfaatkan dalam tahap atau fase eksplorasi ini (*search for prey*). Pada

kenyataannya, para ikan paus bungkuk mencari target/mangsa secara *random* dengan memperhatikan posisi satu dengan yang lain. Oleh karena itu, vektor variasi \vec{A} dengan nilai *random* yaitu lebih dari 1 atau kurang dari -1 digunakan untuk membuat *search agent* bergerak menjauhi ikan paus acuan. Berbeda dengan fase eksploitasi, pada fase eksplorasi posisi *search agent* ditingkatkan berdasarkan posisi *search agent* lain yang dipilih secara *random* atau acak, bukan berdasarkan *search agent* terbaik yang telah ditemukan. Pada fase eksplorasi ini, apabila $|\vec{A}| > 1$ maka algoritma WOA akan diizinkan untuk melakukan pencarian global (Gharehchopogh & Gholizadeh, 2019). Model matematisnya ditunjukkan pada persamaan (2.47) berikut:

$$\vec{D} = |\vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X}| \quad \dots (2.46)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \quad \dots (2.47)$$

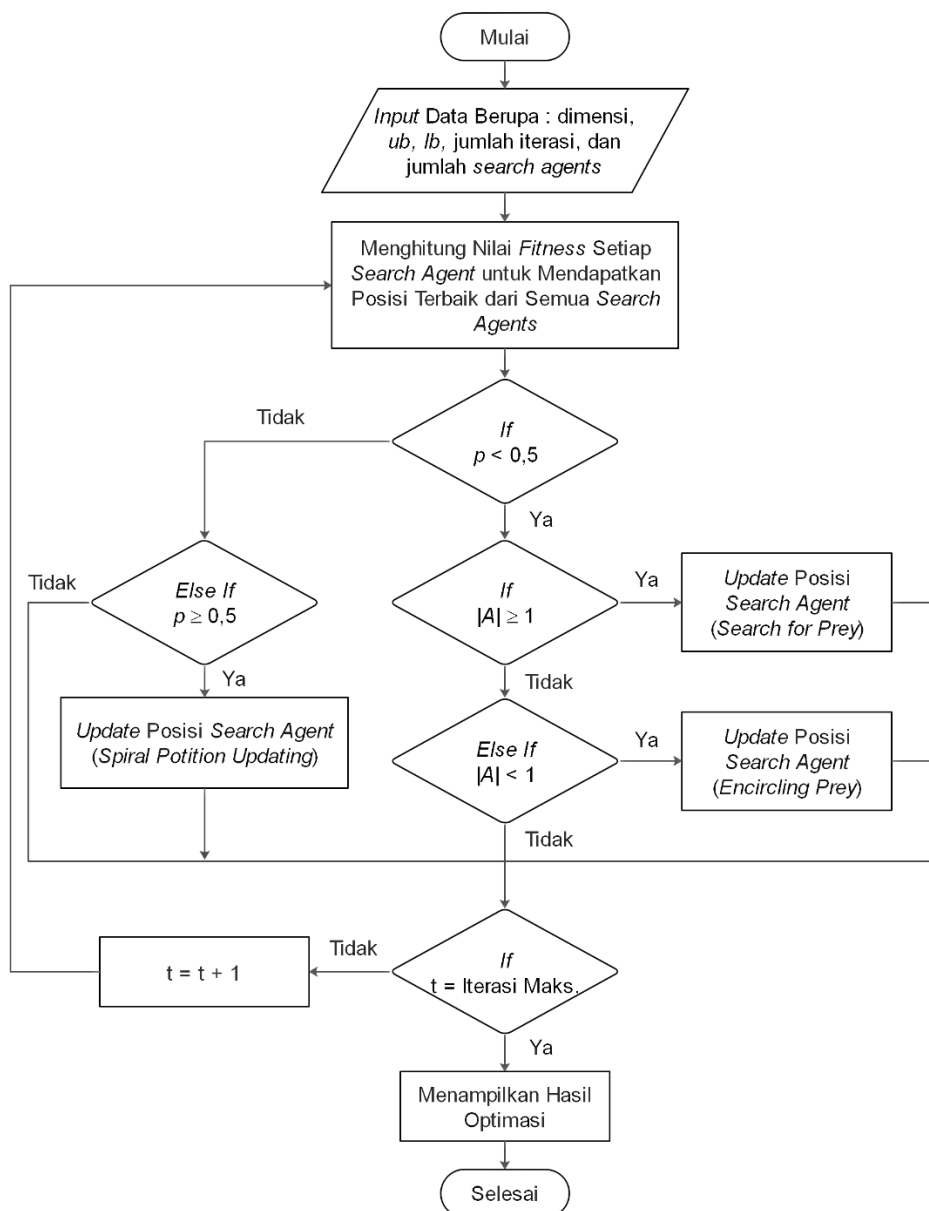
dimana : $\overrightarrow{X_{rand}}$ = vektor posisi *random* yang dipilih sesuai jumlah populasi ikan



Gambar 2.17 Fase Eksplorasi pada WOA (Gharehchopogh & Gholizadeh, 2019)

Algoritma WOA atau *whale optimization algorithm* dimulai dengan mendapatkan nilai optimum secara *random* atau acak. Pada setiap iterasi, *search agent* akan bergerak mendekati posisi *search agent* lain yang terpilih secara *random*

atau posisi terbaik yang didapat sejauh iterasi dilakukan. Parameter a berkurang secara linier dari 2 ke 0 pada setiap iterasi dengan tujuan untuk melakukan fase eksploitasi dan eksplorasi. *Search agent* akan dipilih secara acak apabila $|\vec{A}| > 1$, sedangkan solusi terbaik akan ditentukan ketika $|\vec{A}| < 1$ untuk memperbarui posisi dari *search agent*. Dengan bergantung pada nilai p , algoritma WOA memiliki kemampuan menukar antara pergerakan spiral atau pergerakan melingkar (Gharehchopogh & Gholizadeh, 2019). *Flowchart* atau diagram alir dari proses algoritma WOA secara umum ditunjukkan pada **Gambar 2.18** berikut:



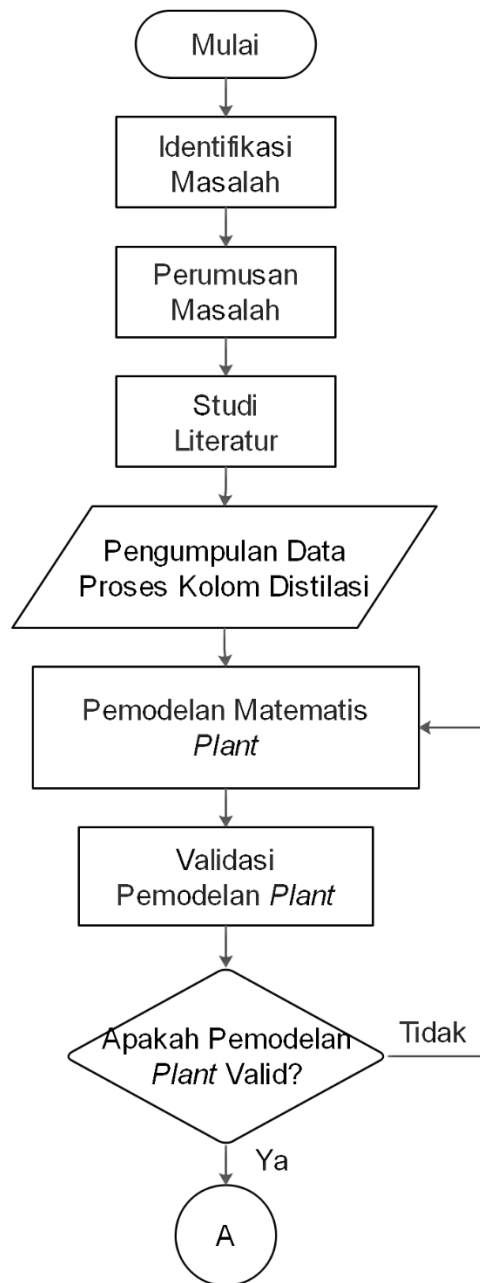
Gambar 2.18 Flowchart Algoritma WOA

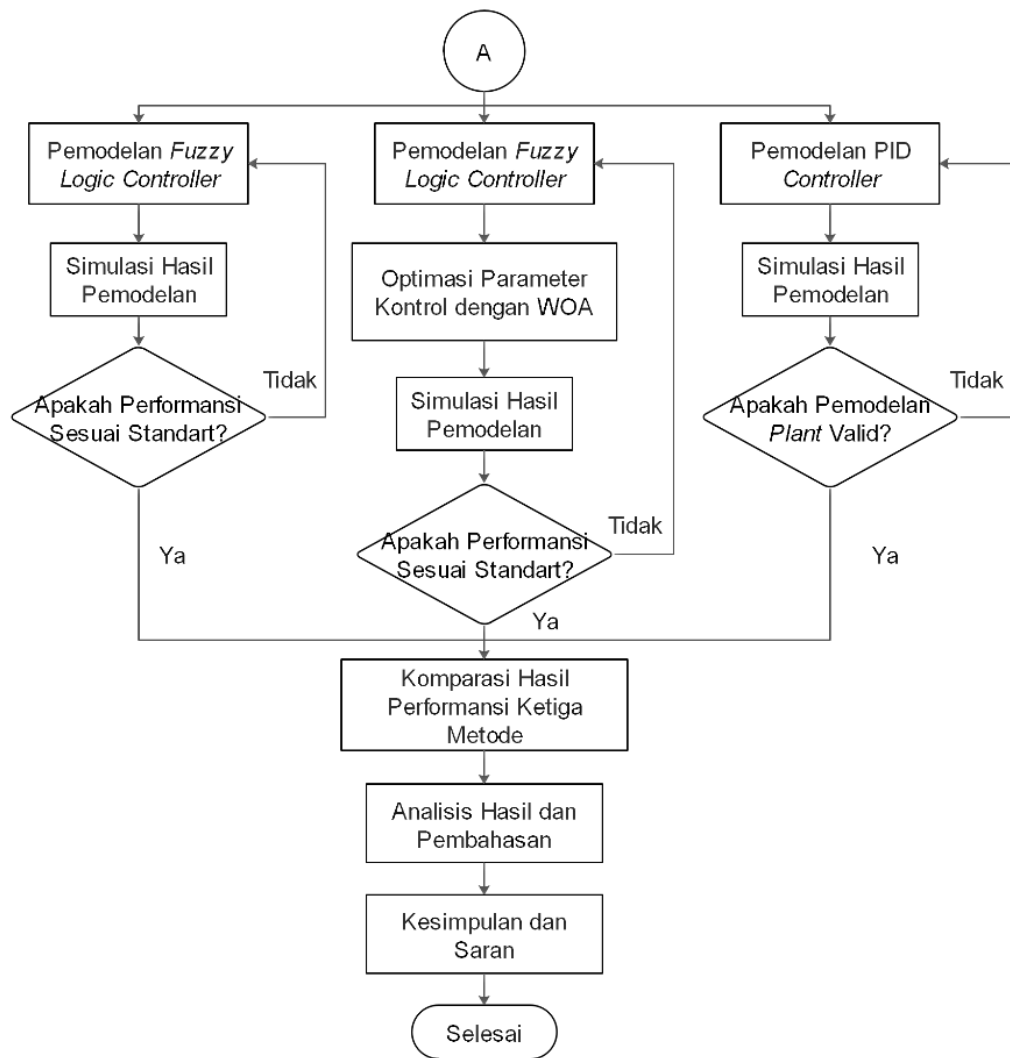
Halaman ini sengaja dikosongkan

BAB III

METODOLOGI PENELITIAN

Diagram alir dari penelitian pada tugas akhir mengenai “Analisis Performansi Sistem Pengendalian pada Kolom Distilasi Menggunakan Fuzzy-*Whale Optimization Algorithm* (Fuzzy-WOA)” ditunjukkan pada **Gambar 3.1** berikut:





Gambar 3.1 Diagram Alir Penelitian

Penjelasan secara rinci mengenai tahapan-tahapan dalam pengerjaan tugas akhir ini sebagai berikut:

3.1 Pengumpulan Data Proses pada Kolom Distilasi

Data yang digunakan pada tugas akhir ini merupakan data sekunder yang diambil dari sebuah jurnal milik (Wood & Berry, 1973). Data tersebut digunakan untuk memodelkan fungsi transfer dari *plant* kolom distilasi (Wood & Berry, 1973). Data yang terdapat pada jurnal (Wood & Berry, 1973) adalah data dalam keadaan *steady*, yaitu berupa *mass flow rate*/laju aliran massa, komposisi, dan suhu. Untuk jenis kolom distilasi yang terdapat pada jurnal (Wood & Berry, 1973) adalah jenis

kolom distilasi biner methanol-air, dimana produk distilatnya adalah methanol dan produk bawahnya adalah air. Data-data yang digunakan untuk memodelkan fungsi transfer kolom distilasi ditunjukkan pada **Tabel 3.1** dan **Tabel 3.2** berikut:

Tabel 3.1 Data *Flow* dan Komposisi pada Kondisi *Steady* (Wood & Berry, 1973)

<i>Stream</i>	<i>Mass Flow Rate</i> [lb/min]	<i>Composition</i> [wt.% methanol]
<i>Overhead</i>	1.18	96.0
<i>Reflux</i>	1.95	96.0
<i>Bottom</i>	1.27	0.5
<i>Feed</i>	2.45	46.5
<i>Steam</i>	1.71	-

Tabel 3.2 Data Suhu pada Kondisi *Steady* (Wood & Berry, 1973)

<i>Stream</i>	<i>Suhu [°F]</i>
<i>Reflux</i>	151.7
<i>Feed</i>	168.0
<i>Steam</i>	233.0
<i>Condensate</i>	227.5
<i>Reboiler</i>	209.6
<i>Plate 1</i>	203.6
<i>Plate 2</i>	194.4
<i>Plate 3</i>	181.2
<i>Plate 4</i>	172.9
<i>Plate 5</i>	164.1
<i>Plate 6</i>	156.8
<i>Plate 7</i>	152.1
<i>Plate 8</i>	148.5
<i>Condenser</i>	143.9

3.2 Pemodelan Matematis dan Uji *Open Loop* Kolom Distilasi

3.2.1 Pemodelan Matematis Kolom Distilasi

Pemodelan kolom distilasi diawali dengan menentukan variabel yang dikendalikan dan variabel yang dimanipulasi. Pada tugas akhir ini, variabel yang dikendalikan adalah komposisi produk distilat dan komposisi produk bawah sedangkan variabel yang dimanipulasi adalah laju aliran massa dari refluks dan *steam reboiler*. Hal ini bersesuaian dengan struktur pengendali L-V pada kolom distilasi biner seperti yang ditunjukkan pada **Gambar 2.5**. Gangguan-gangguan yang terdapat pada sistem diabaikan. Diagram blok *plant* kolom distilasi berdasarkan variabel yang dikendalikan dan variabel yang dimanipulasi ditunjukkan pada **Gambar 3.2** berikut:



Gambar 3.2 Blok Diagram dari *Plant* Kolom Distilasi dengan Dua Variabel *Input* dan Dua Variabel *Output*

Pemodelan fungsi transfer kolom distilasi didapatkan dengan menggunakan pendekatan *First Order Plus Death Time* (FOPDT). Bentuk umum dari FOPDT ditunjukkan pada persamaan (3.1) berikut:

$$G(s) = \frac{y(s)}{u(s)} = \frac{K}{\tau s + 1} e^{-\theta s} \quad \dots (3.1)$$

dimana : K = *gain* statis proses

τ = konstanta waktu tunak

θ = *death time*

Terdapat dua pasang (empat buah) fungsi transfer pada sistem ini, hal ini dikarenakan jumlah dari variabel yang dikendalikan dan dimanipulasi adalah

empat, yaitu komposisi distilat dan komposisi bawah sebagai variabel yang dikendalikan, serta laju aliran massa refluks dan laju aliran massa *steam* sebagai variabel yang dimanipulasi. Laju aliran massa refluks akan mempengaruhi komposisi dari produk distilat maupun produk bawah, begitupun juga dengan laju aliran massa *steam*. Fungsi transfer kolom distilasi dalam bentuk FOPDT yang didapat dari jurnal (Wood & Berry, 1973) pada sistem pengendalian komposisi ini ditunjukkan pada persamaan (3.2) sampai dengan (3.5).

$$\frac{x_D}{L} = \frac{12,8e^{-1s}}{16,7s+1} \quad \dots (3.2)$$

$$\frac{x_D}{Qr} = \frac{-18,9e^{-3s}}{21s+1} \quad \dots (3.3)$$

$$\frac{x_B}{L} = \frac{6,6e^{-7s}}{10,9s+1} \quad \dots (3.4)$$

$$\frac{x_B}{Qr} = \frac{-19,4e^{-3s}}{14,4s+1} \quad \dots (3.5)$$

dimana : x_D = komposisi distilat

x_B = komposisi bawah

L = laju aliran refluks

Qr = laju aliran *steam*

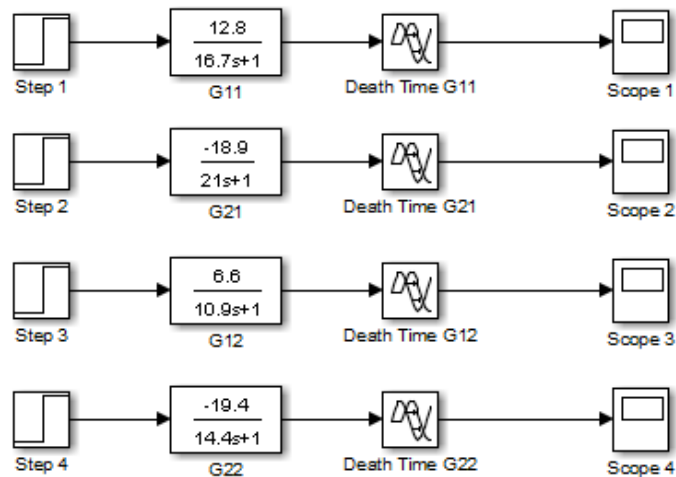
Fungsi transfer dari kolom distilasi biner methanol-air (Wood & Berry, 1973) yang dituliskan dalam bentuk matriks dan menggambarkan hubungan *input* dan *output plant* ditunjukkan pada persamaan (3.6) berikut:

$$\begin{bmatrix} x_D(s) \\ x_B(s) \end{bmatrix} = \begin{bmatrix} \frac{12,8e^{-1s}}{16,7s+1} & \frac{-18,9e^{-3s}}{21s+1} \\ \frac{6,6e^{-7s}}{10,9s+1} & \frac{-19,4e^{-3s}}{14,4s+1} \end{bmatrix} \begin{bmatrix} L(s) \\ Qr(s) \end{bmatrix} \quad \dots (3.6)$$

Fungsi transfer kolom distilasi pada persamaan (3.6) telah divalidasi oleh (Wood & Berry, 1973). Proses validasi dilakukan dengan memberikan sinyal pulsa pada model fungsi transfer *plant*, kemudian hasilnya dibandingkan dengan keadaan asli *plant*. Sinyal pulsa yang diberikan pada *plant* merepresentasikan perubahan laju aliran refluks dan perubahan laju aliran *steam* pada keadaan *real plant*.

3.2.2 Pengujian Respons Step pada Sistem Open Loop

Uji respons *step* sistem *open loop* dilakukan untuk mengetahui karakteristik respons sistem saat kondisi *loop* terbuka apabila diberikan *input* berupa fungsi *step*. Sistem pada tugas akhir ini menggunakan pendekatan model orde satu FOPDT dan *final value* sinyal *step* yang diberikan sebesar 1. Sehingga, respons yang dihasilkan haruslah membentuk kurva eksponensial dan memiliki nilai optimum sesuai dengan nilai *gain*-nya. Uji *open loop* dilakukan pada setiap fungsi transfer *plant* dengan menggunakan *software* Matlab pada fitur *simulink*-nya. Gambar rancangan pada uji *open loop* sistem ditunjukkan pada **Gambar 3.3** berikut:



Gambar 3.3 Uji Respons Step Sistem Open Loop

3.3 Perancangan Sistem Pengendali

Perancangan sistem pengendali pada penelitian ini terdiri dari tiga metode, yaitu sistem pengendali konvensional PID, sistem pengendali dengan sistem logika fuzzy tanpa optimasi, dan sistem pengendali dengan sistem logika fuzzy-whale optimization algorithm (fuzzy-WOA). Pada tugas akhir ini, sistem pengendalian yang dilakukan termasuk jenis sistem pengendalian *multiple input multiple output* (MIMO) karena pada sistem pengendalian ini terdapat dua variabel yang dikontrol dan dua variabel yang dimanipulasi. Model sistem pengendalian MIMO yang digunakan pada tugas akhir ini adalah model sistem pengendalian MIMO desentralisasi 2x2. Sebelum dilakukannya perancangan sistem pengendali, dilakukan perhitungan *relative gain array* (RGA) untuk mengetahui pasangan *input-output plant* (pasangan variabel proses dan variabel dimanipulasi) yang tepat

pada sistem. Perhitungan untuk mendapatkan RGA ditunjukkan pada persamaan (3.7).

$$RGA = \begin{bmatrix} \lambda_{11} & 1 - \lambda_{11} \\ 1 - \lambda_{11} & \lambda_{11} \end{bmatrix} \quad \dots (3.7)$$

Matriks RGA di atas adalah matriks yang akan digunakan untuk menentukan pasangan *input* dan *output* sistem. Nilai λ_{11} didapatkan dengan menggunakan persamaan (2.14). K_{11} , K_{12} , K_{21} , dan K_{22} yang merupakan nilai *gain* dari masing-masing fungsi transfer disubstitusikan ke dalam persamaan (2.14). Didapat nilai λ_{11} seperti yang ditunjukkan pada persamaan (3.9).

$$\lambda_{11} = \frac{1}{1 - \left[\frac{(-18.9) \cdot 6.6}{12.8 \cdot (-19.4)} \right]} \quad \dots (3.8)$$

$$\lambda_{11} = 2.00803 \quad \dots (3.9)$$

$$1 - \lambda_{11} = -1.00803 \quad \dots (3.10)$$

Dari perhitungan di atas, nilai λ_{11} yang didapat disubstitusikan ke dalam matriks (3.7), sehingga didapat matriks RGA seperti pada persamaan (3.11) berikut:

$$RGA = \begin{bmatrix} 2.00803 & -1.00803 \\ -1.00803 & 2.00803 \end{bmatrix} \quad \dots (3.11)$$

Untuk memudahkan analisis dari hasil perhitungan RGA, persamaan (3.11) diubah ke dalam bentuk tabel. Hasil perhitungan RGA dalam bentuk tabel ditunjukkan pada **Tabel 3.3** berikut ini:

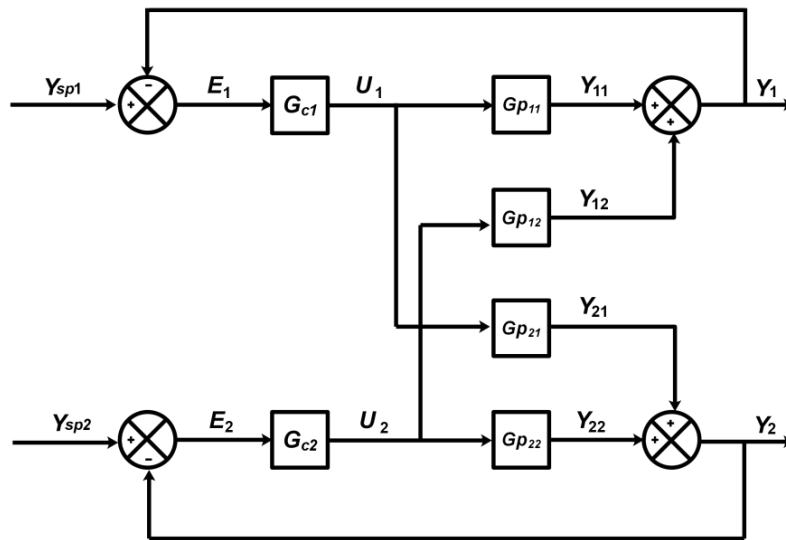
Tabel 3.3 Analisis Pasangan *Input* dan *Output*

<i>Input/Output</i>	U1	U2
Y1	2.00803	-1.00803
Y2	-1.00803	2.00803

Pasangan *input-output* yang baik ketika merancang sebuah sistem pengendalian MIMO adalah pasangan yang tidak menyebabkan adanya interaksi. Aturan RGA *pairing* mengatakan bahwa nilai RGA yang dipilih tidak boleh negatif

dan harus mendekati nilai 1. Nilai RGA negatif menandakan bahwa pasangan *input-output* apabila dipasangkan akan menghasilkan nilai yang berkebalikan (*inverse* antara perubahan *input* dan perubahan *output*), sedangkan apabila RGA bernilai 1 menandakan bahwa pasangan *input-output* tidak menyebabkan adanya interaksi. Nilai RGA yang lebih dari 1 menandakan bahwa pasangan *input-output* akan menyebabkan adanya interaksi. Interaksi akan semakin besar (menyebabkan *plant* semakin tidak stabil) apabila nilai RGA yang dihasilkan semakin besar dan jauh dari nilai 1 (Yang, 2020).

Pasangan $Y_1;U_1$ dan pasangan $Y_2;U_2$ memiliki nilai positif yaitu sebesar 2,00803 seperti yang ditunjukkan pada **Tabel 3.3**, sehingga pasangan variabel proses dan variabel dimanipulasi yang tepat pada sistem pengendalian komposisi ini adalah $Y_1;U_1$ dan $Y_2;U_2$. Langkah selanjutnya setelah mendapatkan pasangan *input* dan *output* sistem yang tepat adalah dilakukanlah perancangan sistem pengendali. Struktur rancangan sistem pengendali ditunjukkan melalui diagram blok pada **Gambar 3.4** berikut:



Gambar 3.4 Diagram Blok Sistem Pengendalian MIMO Desentralisasi 2X2
(Sutapa et al., 2017)

dimana : Y_{sp1} = *set point* komposisi distilat

Y_{sp2} = *set point* komposisi bawah

E_1 = *error* komposisi distilat

E_2 = error komposisi bawah

G_{c1} = pengendali pada komposisi distilat

G_{c2} = pengendali pada komposisi bawah

U_1 = output pengendali pada komposisi distilat

U_2 = output pengendali pada komposisi bawah

G_p = fungsi transfer kolom distilasi

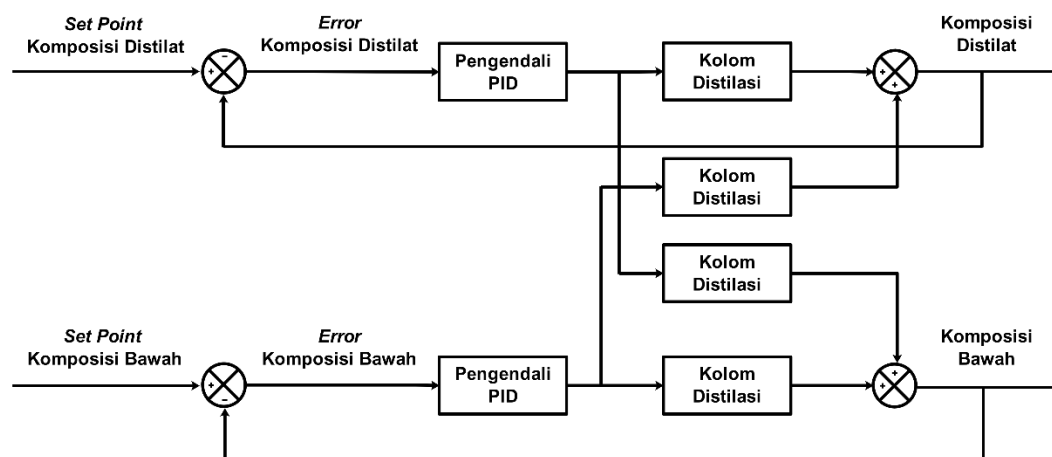
Y_1 = komposisi distilat

Y_2 = komposisi bawah

G_{p11} , G_{p12} , G_{p21} , dan G_{p22} , pada **Gambar 3.4** secara berturut-turut adalah fungsi transfer kolom distilasi biner methanol-air pada persamaan (3.2), (3.3), (3.4), dan (3.5). Untuk blok pengendali G_c pada **Gambar 3.4** berisi pengendali yang dirancang pada penelitian ini, yaitu pengendali PID, fuzzy tanpa optimasi, dan fuzzy-WOA. Nilai *set point* yang diberikan pada sistem disesuaikan dengan data yang didapat pada jurnal (Wood & Berry, 1973), yaitu pada komposisi distilat sebesar 96% dan pada komposisi bawah sebesar 0,5%.

3.3.1 Perancangan Pengendali Konvensional PID

Perancangan sistem pengendalian komposisi pada kolom distilasi dengan menggunakan pengendali PID ditunjukkan pada **Gambar 3.5** berikut:



Gambar 3.5 Diagram Blok Sistem Pengendalian Komposisi pada Kolom Distilasi dengan Pengendali PID

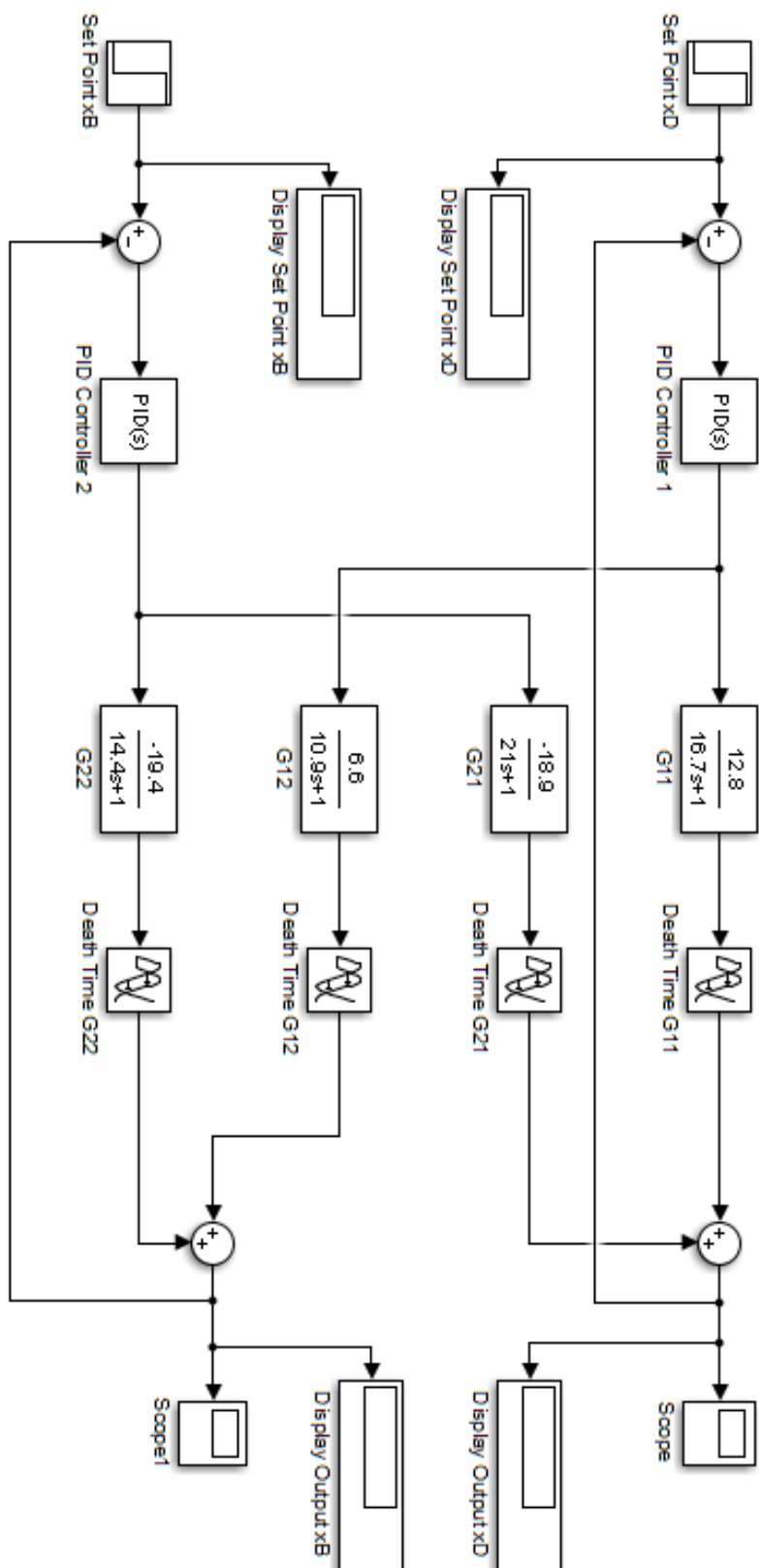
Langkah awal dalam merancang sistem pengendalian komposisi pada kolom distilasi menggunakan pengendali PID adalah proses *tuning* atau proses penentuan parameter pengendali. Parameter pengendali PID akan menentukan bagus tidaknya hasil respons suatu sistem pengendalian. Setiap *plant* akan memiliki nilai parameter pengendali PID yang berbeda pula. Pada tugas akhir ini, proses *tuning* parameter pengendali PID dilakukan dengan metode *autotune* pada *toolbox* PID controller *simulink*. Pengendali yang digunakan pada sistem pengendalian komposisi ini adalah pengendali tipe-P, tipe-PI, dan tipe-PID. Ketiga pengendali dituning satu persatu secara bergantian mulai dari tipe-P, tipe-PI, lalu tipe-PID. Pengendali yang digunakan pada sistem ini berjumlah dua seperti yang ditunjukkan pada **Gambar 3.5**, hal ini bersesuaian dengan struktur diagram blok sistem kendali MIMO desentralisasi 2x2 pada **Gambar 3.4**. Pengendali PID diperlakukan sama pada kasus ini, artinya ketika pengendali 1 (pengendali pada komposisi distilat) menggunakan pengendali tipe-P, maka pengendali 2 (pengendali pada komposisi bawah) juga menggunakan pengendali tipe-P. *Tuning* dilakukan secara bergantian dimulai dari pengendali pertama dilanjut dengan *tuning* pada pengendali kedua. *Tuning* dilakukan secara terus-menerus hingga simulasi respons dinamik sistem menunjukkan hasil yang terbaik. Didapat parameter pengendali dari hasil *tuning* menggunakan Matlab yang ditunjukkan pada **Tabel 3.4** dan **Tabel 3.5** berikut:

Tabel 3.4 Parameter PID pada Pengendali 1

Pengendali	Kp	Ki	Kd	Filter Coefficient
P	0.1396	-	-	-
PI	0.1517	0.0177	-	-
PID	0.3346	0.0337	0.0576	0.2809

Tabel 3.5 Parameter PID pada Pengendali 2

Pengendali	Kp	Ki	Kd	Filter Coefficient
P	-0.0777	-	-	-
PI	-0.0197	-0.0092	-	-
PID	-0.0270	-0.0120	0	100

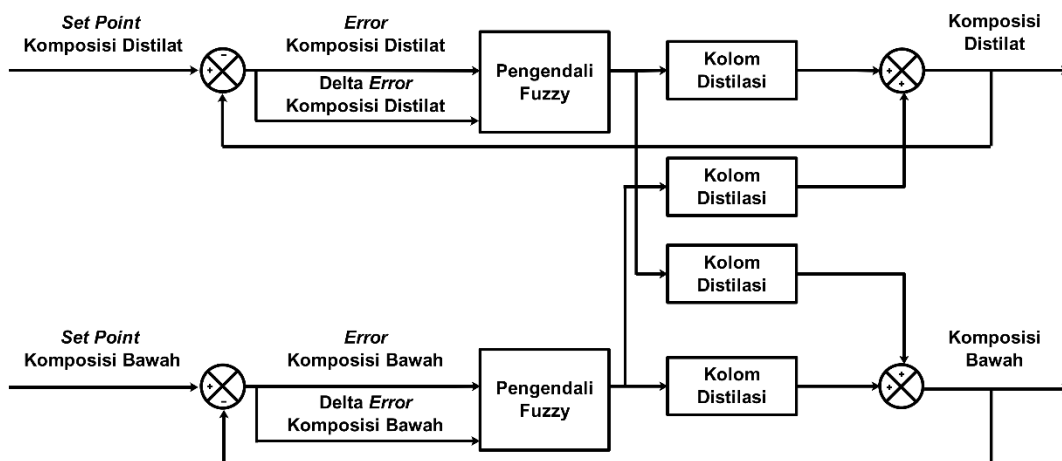


Gambar 3.6 Simulasi Pengendali PID pada Simulink

Simulasi kemudian dilakukan untuk mengetahui performansi sistem akibat pengendali PID yang telah dirancang setelah parameter PID dari masing-masing tipe pengendali berhasil didapat. **Gambar 3.6** adalah gambar simulasi sistem pengendalian komposisi pada kolom distilasi berbasis PID pada *simulink* Matlab.

3.3.2 Perancangan Pengendali Fuzzy

Perancangan sistem pengendalian komposisi pada kolom distilasi dengan menggunakan pengendali fuzzy tanpa optimasi ditunjukkan melalui diagram blok pada **Gambar 3.7** di bawah ini:



Gambar 3.7 Diagram Blok Sistem Pengendalian Komposisi pada Kolom Distilasi dengan Pengendali Fuzzy tanpa Optimasi

Perancangan dimulai dengan menentukan rancangan fungsi keanggotaan *input* dan juga *output*. *Input* pengendali fuzzy pada sistem ini berupa *error* dan *delta error* komposisi, sedangkan *output* berupa sinyal kontrol. Jumlah himpunan fuzzy yang digunakan pada sistem ini berjumlah lima, yaitu *negative large* (NL), *negative small* (NS), *zero* (ZE), *positive small* (PS), dan *positive large* (PL). Hal tersebut berlaku pada semua fungsi keanggotaan baik pada fungsi keanggotaan *input* maupun *output*. Ada dua bentuk fungsi keanggotaan yang digunakan pada sistem pengendalian komposisi ini, yaitu fungsi keanggotaan bahu dan segitiga. Bentuk fungsi keanggotaan didesain sama dalam satu sistem pengendali fuzzy. Sehingga, akan ada dua buah skenario perancangan sistem pengendali fuzzy, yaitu sistem

pengendali fuzzy dengan fungsi keanggotaan *input* maupun *output* berbentuk bahu dan sistem pengendali fuzzy dengan fungsi keanggotaan *input* maupun *output* berbentuk segitiga. Gambaran skenario rancangan pengendali fuzzy ditunjukkan pada **Tabel 3.6** berikut:

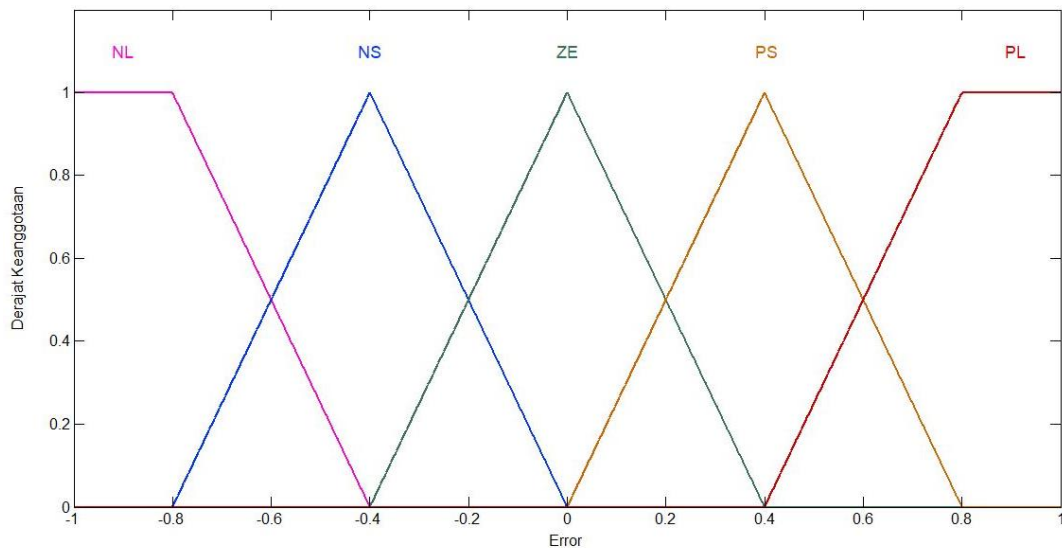
Tabel 3.6 Skenario Sistem Pengendali Fuzzy

Skenario	Bentuk Fungsi Keanggotaan	Variabel Fuzzy	Jumlah Himpunan Fuzzy
Skenario 1	Bahu	Error (<i>input</i>), Delta Error (<i>input</i>), dan Sinyal Kontrol (<i>output</i>)	5 (NL, NS, ZE, PS, PL)
Skenario 2	Segitiga	Error (<i>input</i>), Delta Error (<i>input</i>), dan Sinyal Kontrol (<i>output</i>)	5 (NL, NS, ZE, PS, PL)

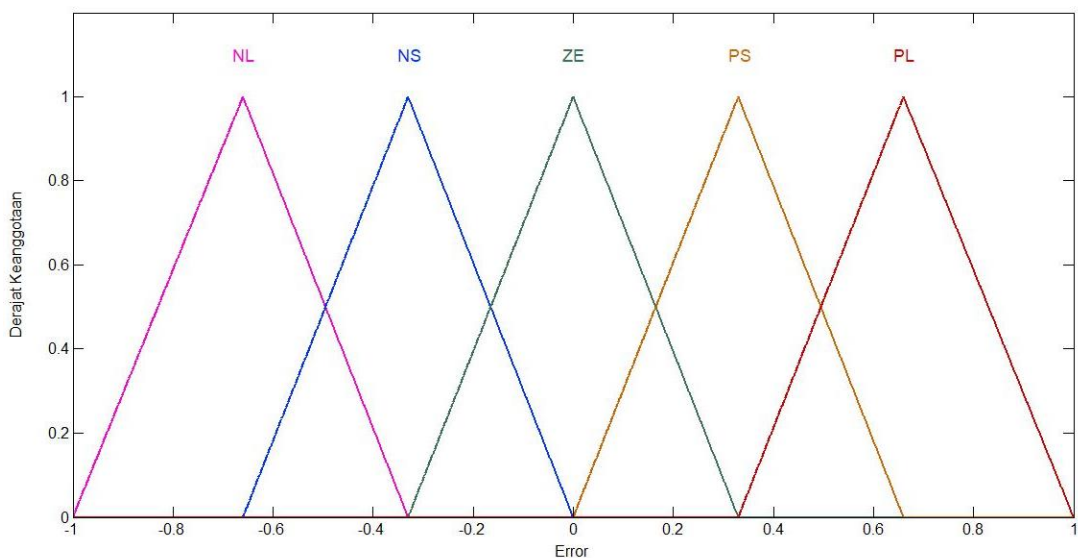
Dirancangnya kedua skenario sistem pengendali fuzzy seperti yang ditunjukkan pada **Tabel 3.6** dengan tujuan untuk mengetahui bentuk fungsi keanggotaan yang mana yang akan menghasilkan respons dinamik dan performansi yang paling baik. Sistem pengendali fuzzy yang memiliki respons dinamik dan performansi terbaik, akan dibandingkan dengan hasil performansi pengendali PID dan fuzzy-WOA yang terbaik juga. Untuk mendapatkan hasil respons dan performansi yang baik, *range* pada fungsi keanggotaan *input-output* dan juga *rule base* dirancang sedemikian rupa.

Range pada fungsi keanggotaan *input error* dirancang bernilai -1 sampai dengan 1. *Range* -1 sampai dengan 1 didapatkan dengan memperhitungkan nilai *error* minimum dan maksimum yang mungkin terjadi pada sistem. Lebih jelasnya, *range* didapatkan berpatokan pada nilai komposisi yang memiliki nilai minimum 0% dan nilai maksimum 100%. Nilai *range* ini berlaku pada kedua sistem pengendali fuzzy, baik pada sistem pengendali fuzzy dengan fungsi keanggotaan

input error berbentuk bahu maupun pada sistem pengendali fuzzy dengan fungsi keanggotaan *input error* berbentuk segitiga. Fungsi keanggotaan dari *input error* pada kedua sistem pengendali fuzzy ditunjukkan pada **Gambar 3.8** dan **Gambar 3.9** berikut:



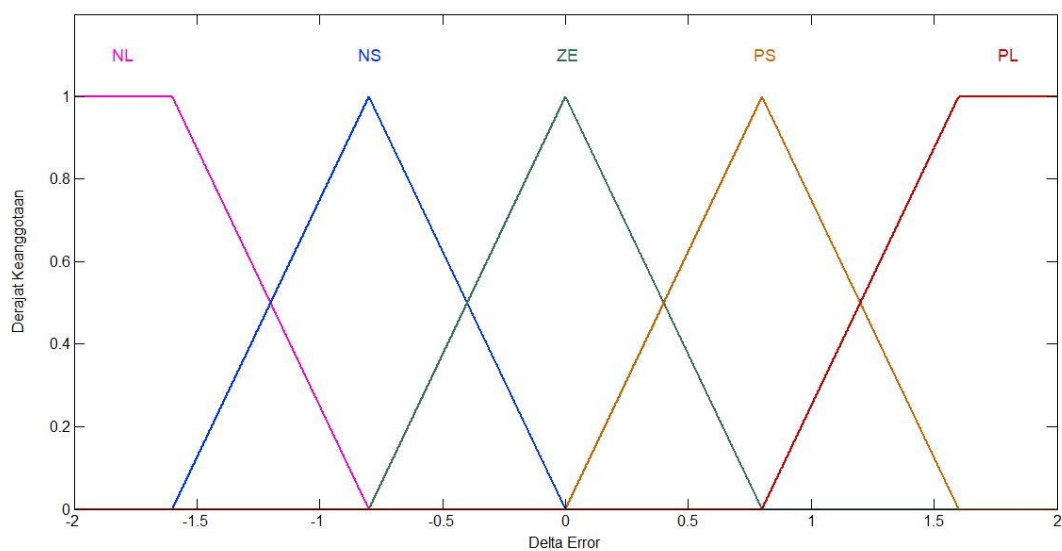
Gambar 3.8 *Input Error* dengan Fungsi Keanggotaan Bentuk Bahu



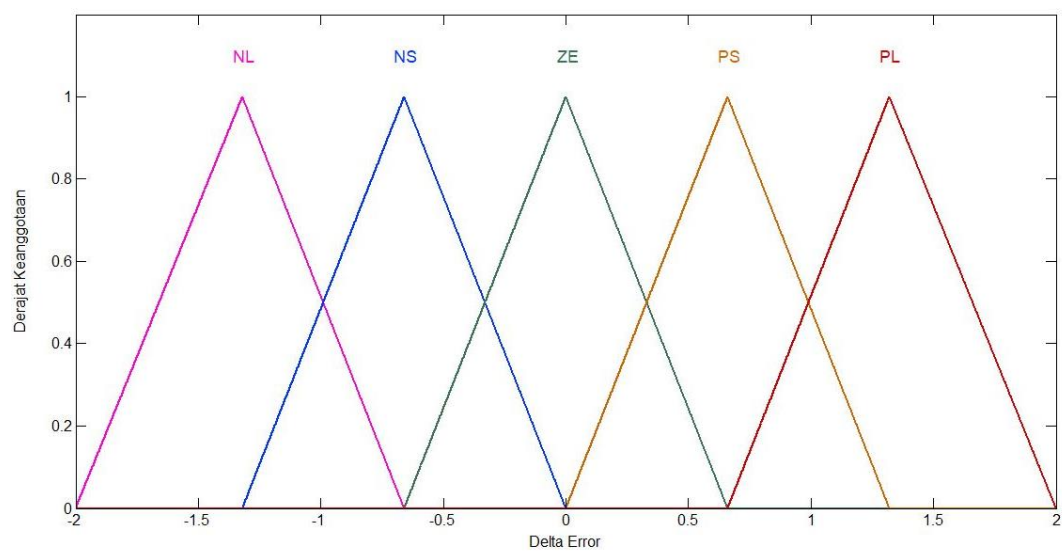
Gambar 3.9 *Input Error* dengan Fungsi Keanggotaan Bentuk Segitiga

Range pada fungsi keanggotaan *input delta error* dirancang bernilai -2 sampai dengan 2. *Range* dari *input delta error* ini didapatkan berdasarkan kemungkinan

yang pasti terjadi pada sistem, dengan menggunakan nilai minimum dan nilai maksimum *error* sebagai patokan, yaitu -1 dan 1. Delta *error* dirumuskan sebagai hasil pengurangan antara *error* saat ini dengan *error* sebelumnya, sehingga nilai minimum dan maksimum yang mungkin dihasilkan adalah -2 dan 2. Nilai *range* ini juga berlaku pada kedua sistem pengendali fuzzy. Fungsi keanggotaan dari *input* delta *error* pada kedua sistem pengendali fuzzy ditunjukkan pada **Gambar 3.10** dan **Gambar 3.11** berikut:

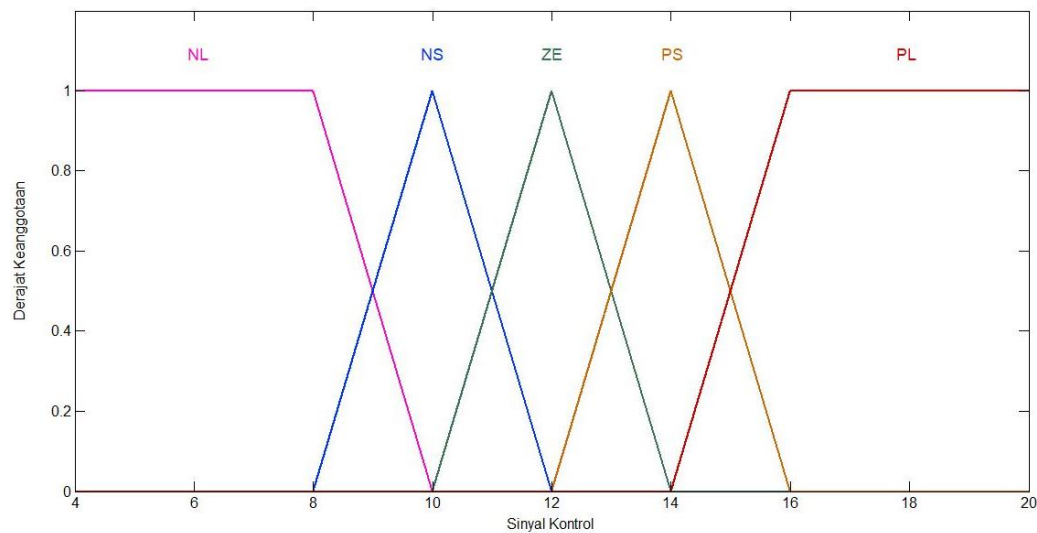


Gambar 3.10 *Input Delta Error* dengan Fungsi Keanggotaan Bentuk Bahu

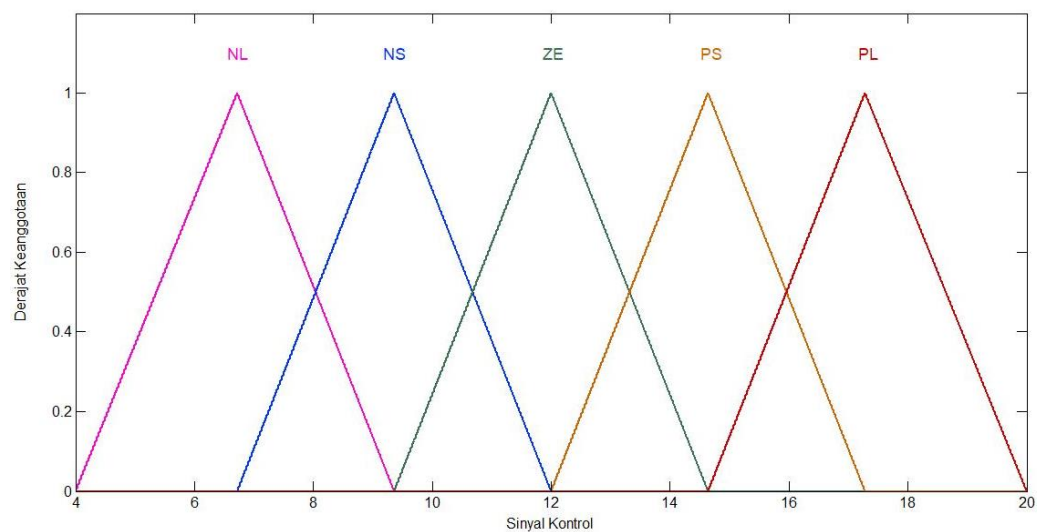


Gambar 3.11 *Input Delta Error* dengan Fungsi Keanggotaan Bentuk Segitiga

Terakhir, *range* pada fungsi keanggotaan *output* dirancang memiliki nilai 4 sampai dengan 20. Hal ini bersesuaian dengan keluaran sinyal kontrol secara umum, yaitu berupa arus 4-20mA sebagai masukan pada *valve*. Sama halnya dengan *input error* dan *input delta error*, *range* ini berlaku pada kedua sistem pengendali fuzzy. Fungsi keanggotaan *output* pada kedua sistem pengendali fuzzy ditunjukkan pada **Gambar 3.12** dan **Gambar 3.13** berikut:



Gambar 3.12 Output Sistem dengan Fungsi Keanggotaan Bentuk Bahu



Gambar 3.13 Output Sistem dengan Fungsi Keanggotaan Bentuk Segitiga

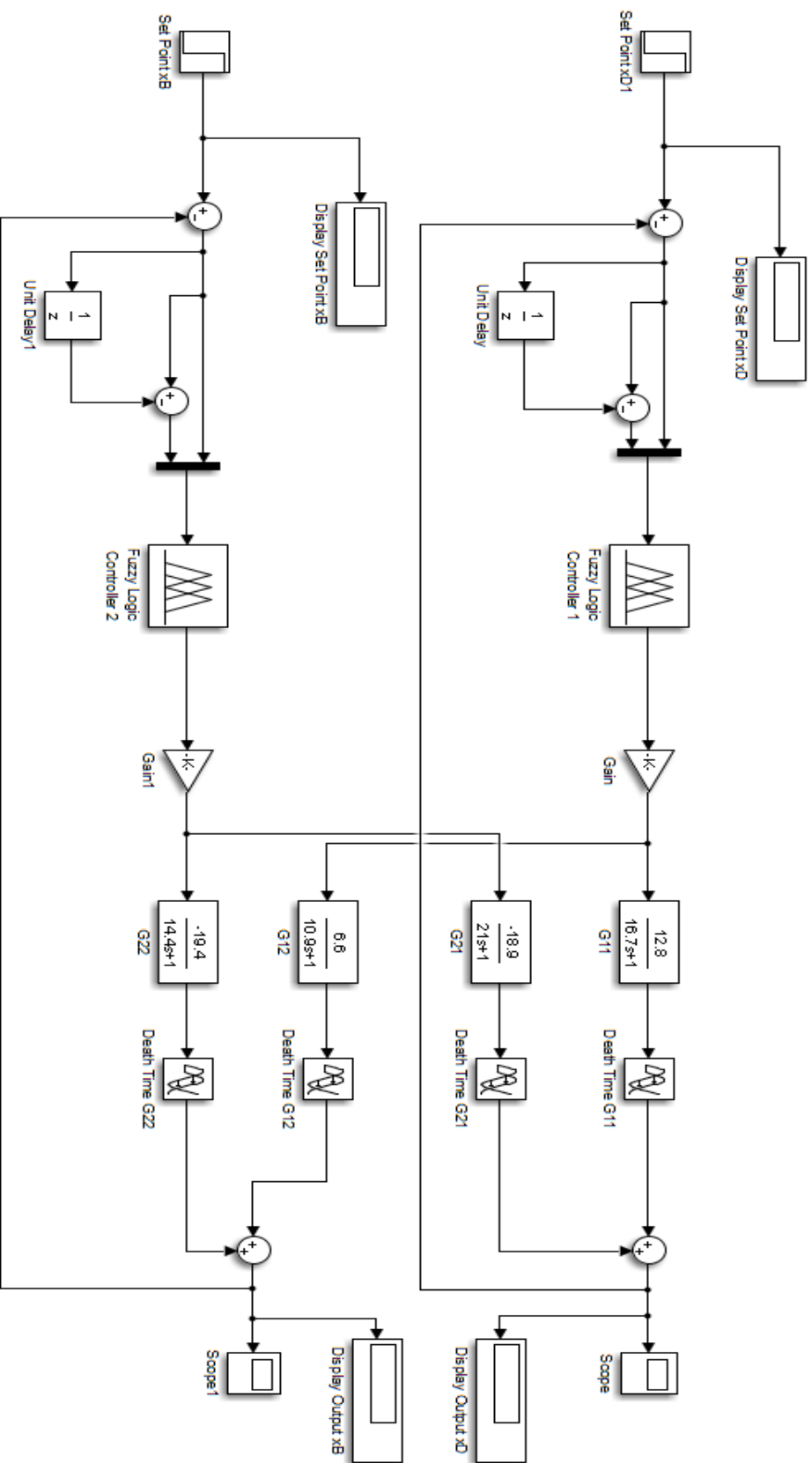
Setelah fungsi keanggotaan *input* dan *output* didapat, dilakukanlah perancangan terhadap *rule base*. Jumlah *rule base* suatu sistem logika fuzzy

bergantung pada jumlah *input* dan juga jumlah himpunan fuzzy setiap *input*. Untuk sistem fuzzy yang memiliki dua *input*, jumlah *rule base* didapat dengan saling mengalikan jumlah himpunan fuzzy pada masing masing *input* tersebut. Pada sistem pengendalian ini himpunan fuzzy dari kedua *input*, masing-masing berjumlah lima sehingga didapat jumlah *rule base* adalah sebanyak 25. Jenis sistem inferensi yang digunakan pada sistem ini adalah penalaran Mamdani. Operator yang digunakan adalah operator AND, sedangkan untuk metode yang digunakan pada proses defuzzifikasi adalah metode *centroid*. Desain *rule base* pada sistem pengendalian komposisi ditunjukkan pada **Tabel 3.7** berikut:

Tabel 3.7 Tabel *Rule Base* Sistem

Error / Delta Error	NL	NS	ZE	PS	PL
NL	NL	NL	NS	NS	ZE
NS	NL	NS	NS	ZE	PS
ZE	NS	NS	ZE	PS	PS
PS	NS	ZE	PS	PS	PL
PL	ZE	PS	PS	PL	PL

Rancangan diagram blok sistem pengendalian komposisi berbasis fuzzy sebagian besar sama dengan diagram blok sistem pengendalian komposisi berbasis PID, hanya saja terdapat penambahan *input* delta *error* terhadap *input* pengendali fuzzy dan *gain* pada keluaran pengendali fuzzy. *Gain* digunakan untuk mengubah keluaran fuzzy yang berupa sinyal kontrol menjadi nilai laju aliran sebagai *input plant*. Sama halnya dengan sistem pengendalian komposisi berbasis PID, sistem pengendali yang diaplikasikan pada sistem utama adalah sama. Apabila pengendali 1 adalah sistem pengendali fuzzy dengan fungsi keanggotaan bahu, maka pengendali 2 juga sistem pengendali fuzzy dengan fungsi keanggotaan bahu dan tentunya dengan parameter yang sama pula.



Gambar 3.14 Simulasi Pengendali Fuzzy pada Simulink

Sistem pengendali fuzzy kemudian diaplikasikan dan disimulasikan pada sistem utama setelah sistem pengendali fuzzy selesai dirancang. **Gambar 3.14** adalah gambar simulasi sistem pengendalian komposisi pada kolom distilasi berbasis fuzzy tanpa optimasi pada *simulink* Matlab.

3.3.3 Perancangan Pengendali Fuzzy-WOA

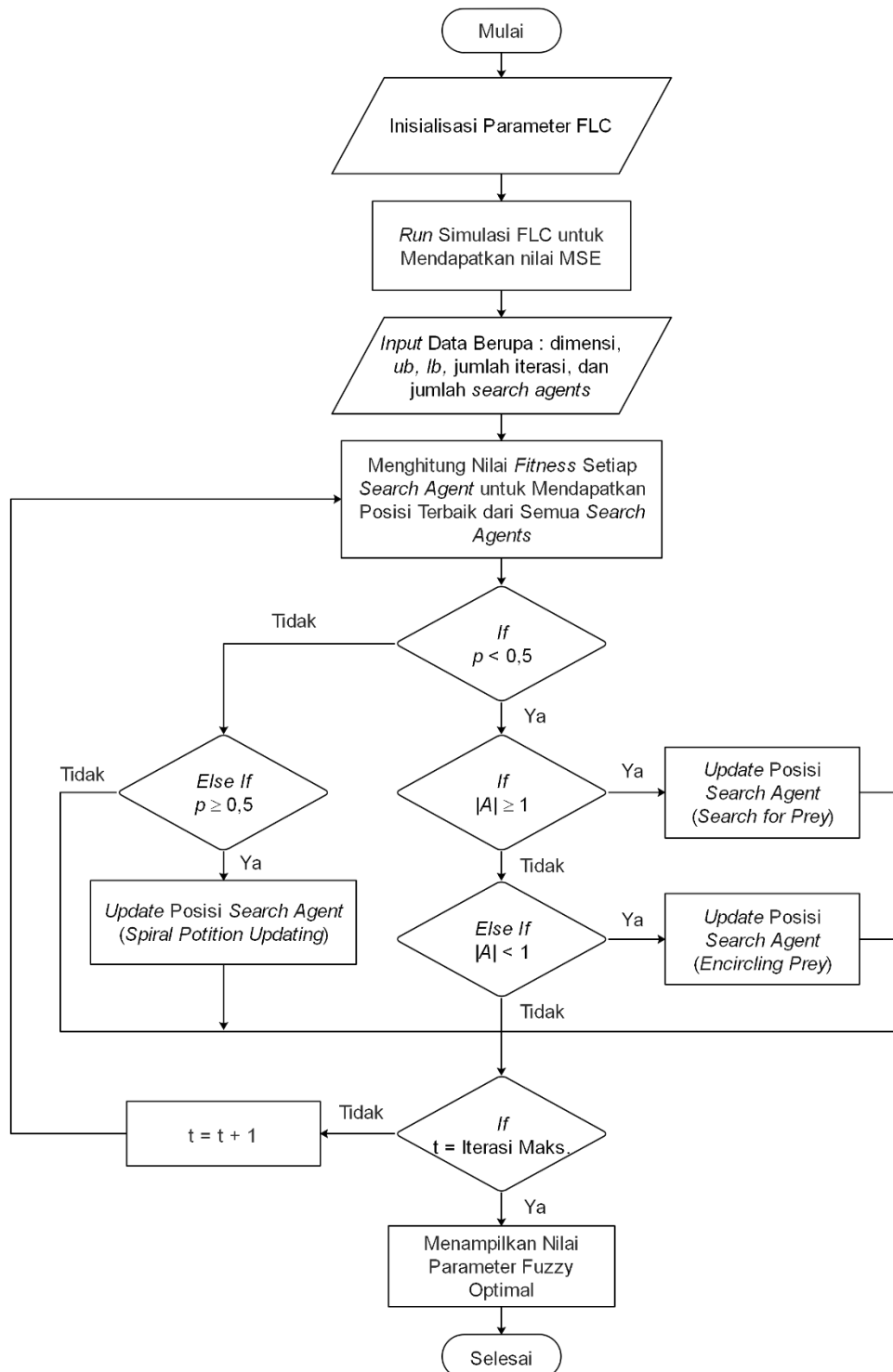
Salah satu kendala ketika merancang sebuah sistem pengendalian berbasis logika fuzzy adalah tidak adanya aturan resmi dan pasti dalam menentukan bentuk fungsi keanggotaan dan juga letak fungsi keanggotaan yang optimal. Maka dari itu, dilakukanlah metode optimasi parameter fuzzy dengan menggunakan *whale optimization algorithm* (WOA). Parameter fuzzy yang dioptimasi dengan WOA berupa *range* dan letak titik pada fungsi keanggotaan *input-output*. Optimasi ini dilakukan untuk mengoptimalkan hasil rancangan sistem pengendali fuzzy tanpa optimasi, yaitu dari segi *range* dan letak titik pada fungsi keanggotaan *input-output*. Perancangan pengendali fuzzy-WOA dilakukan terhadap rancangan pengendali fuzzy tanpa optimasi pada salah satu skenario yang menghasilkan performansi terbaik.

Perancangan pengendali fuzzy-WOA terdiri dari empat skenario, dimana dilakukan variasi nilai pada parameter WOA, yaitu berupa jumlah *search agent* dan jumlah iterasi maksimum. Untuk parameter *ub* dan *lb* disesuaikan dengan rentang nilai setiap himpunan fuzzy tanpa optimasi yang telah dirancang sebelumnya. Skenario perancangan pengendali fuzzy-WOA ditunjukkan pada **Tabel 3.8** berikut:

Tabel 3.8 Skenario Perancangan Fuzzy-WOA

Skenario	Jumlah <i>Search Agent</i>	Nilai Iterasi Maksimum
Skenario 1	100	500
Skenario 2	1000	500
Skenario 3	100	700
Skenario 4	1000	700

Proses optimasi parameter pengendali logika fuzzy/fuzzy logic controller menggunakan *whale optimization algorithm* ditunjukkan melalui diagram alir pada **Gambar 3.15** berikut:



Gambar 3.15 Diagram Alir Optimasi Parameter *Fuzzy Logic Controller*

Penjelasan rinci mengenai proses optimasi parameter pengendali logika fuzzy/fuzzy logic controller menggunakan *whale optimization algorithm* (WOA) pada sistem pengendalian komposisi kolom distilasi (MIMO 2x2) sesuai dengan diagram alir yang ditunjukkan pada **Gambar 3.15** adalah sebagai berikut:

1. Parameter pengendali fuzzy yang dioptimasi dengan WOA adalah parameter fuzzy dari skenario 1 pada pengendali fuzzy tanpa optimasi. Optimasi dilakukan pada ketiga variabel fuzzy, yaitu *input error*, *input delta error*, dan *output* sinyal kontrol yang berturut-turut ditunjukkan pada **Gambar 3.8**, **Gambar 3.10**, dan **Gambar 3.12**.
2. Simulasi pada skenario 1 pengendali fuzzy tanpa optimasi *dirunning* untuk mendapatkan nilai *error* agar dapat menghitung nilai *mean squared error* (MSE) sebagai fungsi *fitness*.
3. Memasukkan parameter WOA berupa dimensi, *ub*, *lb*, jumlah *search agent*, dan jumlah iterasi maksimum. Nilai dimensi disesuaikan dengan jumlah parameter dari setiap bentuk keanggotaan fuzzy, yaitu 3 untuk bentuk segitiga dan 4 untuk bentuk trapesium. Untuk parameter *ub* dan *lb* disesuaikan dengan rentang nilai setiap himpunan fuzzy tanpa optimasi pada skenario 1, sedangkan untuk jumlah *search agent* dan jumlah iterasi maksimum divariasi sesuai **Tabel 3.8**.
4. Pencarian nilai *fitness* terbaik pada setiap *search agent* dilakukan dengan menggunakan nilai MSE, adapun rumus dari MSE ditunjukkan pada persamaan (2.38).
5. Setiap *search agent* akan bergerak mendekati nilai optimum (parameter fuzzy optimum), sehingga posisi setiap *search agent* akan berubah mendekati posisi optimum yang didapat pada iterasi ke *t*. Proses ini akan berlangsung sampai jumlah iterasi maksimum tercapai.
6. Nilai parameter fuzzy optimum didapat dan ditampilkan saat sistem telah mencapai iterasi maksimum.

Hasil yang didapat dari proses optimasi parameter pengendali fuzzy dengan menggunakan *whale optimization algorithm* dituliskan dalam bentuk tabel. Hasil

dari optimasi parameter pengendali fuzzy yang didapat dengan menggunakan *whale optimization algorithm* ditunjukkan pada **Tabel 3.9**, **Tabel 3.10**, dan **Tabel 3.11**.

Tabel 3.9 Hasil Optimasi Parameter Fuzzy pada Variabel *Input Error*

Skenario	Parameter Fuzzy Hasil Optimasi				
	NL	NS	ZE	PS	PL
Skenario 1	[-0.9642; -0.9027; -0.6134 ; -0.5122]	[-0.6660; -0.4556; -0.0366]	[-0.1128; 0.3253; 0.3999]	[0.0089; 0.2495; 0.3865]	[0.4611; 0.6003; 0.7848; 0.8571]
Skenario 2	[-0.8900; -0.8320; -0.5965; -0.4021]	[-0.7369; -0.2083; -0.0658]	[-0.0774; 0.1279; 0.3779]	[0.0313; 0.5623; 0.7360]	[0.4880; 0.6765; 0.8275; 0.9203]
Skenario 3	[-0.9953; -0.9084; -0.7607; -0.6295]	[-0.7386; -0.5861; -0.3283]	[-0.3331; 0.2794; 0.2832]	[0.1514; 0.3039; 0.7767]	[0.4925; 0.5754; 0.6161; 0.7298]
Skenario 4	[-0.9951; -0.5735; -0.5116; -0.4815]	[-0.5693; -0.3528; -0.0012]	[-0.3467; 0.2304; 0.3947]	[0.1418; 0.7104; 0.7382]	[0.6914; 0.7098; 0.8984; 0.9641]

Tabel 3.9 merupakan tabel hasil optimasi parameter fuzzy dengan menggunakan *whale optimization algorithm* pada variabel fuzzy *input error*. Rentang nilai *input error* yang didapat dari hasil optimasi berturut-turut dari

skenario 1, skenario 2, skenario 3, dan skenario 4 adalah (-0.9642 s/d 0.8571), (-0.8900 s/d 0.9203), (-0.9953 s/d 0.7298), dan (-0.9951 s/d 0.9641).

Tabel 3.10 Hasil Optimasi Parameter Fuzzy pada Variabel *Input Delta Error*

Skenario	Parameter Fuzzy Hasil Optimasi				
	NL	NS	ZE	PS	PL
Skenario 1	[-1.8298; -1.6065; -1.3591; -1.0130]	[-1.4324; -1.1351; -0.2164]	[-0.1286; 0.3776; 0.7907]	[0.0209; 1.2889; 1.4468]	[0.8922; 0.9649; 1.4815; 1.8557]
Skenario 2	[-1.7542; -1.3326; -1.0077; -1.0011]	[-1.1140; -0.7233; -0.1933]	[-0.3337; 0.4436; 0.7363]	[0.4677; 1.2088; 1.5759]	[0.8223; 1.3719; 1.8081; 1.8867]
Skenario 3	[-1.8295; -1.5564; -1.4913; -0.9615]	[-1.0698; -0.7217; 0.6580]	[-0.5385; -0.0578; 0.0512]	[0.1487; 1.2122; 1.5216]	[0.8804; 1.1124; 1.4895; 1.6364]
Skenario 4	[-1.7536; -1.6647; -1.4719; -1.0598]	[-1.5940; -0.7763; -0.3249]	[-0.4859; -0.1217; 0.3586]	[0.3056; 1.1368; 1.2222]	[0.8497; 0.9498; 1.7032; 1.8355]

Tabel 3.10 merupakan tabel hasil optimasi parameter fuzzy dengan menggunakan *whale optimization algorithm* pada variabel fuzzy *input delta error*. Rentang nilai *input delta error* yang didapat dari hasil optimasi berturut-turut dari

skenario 1, skenario 2, skenario 3, dan skenario 4 adalah (-1.8298 , 1.8557), (-1.7542 , 1.8867), (-1.8295 , 1.6364), dan (-1.7536 , 1.8355).

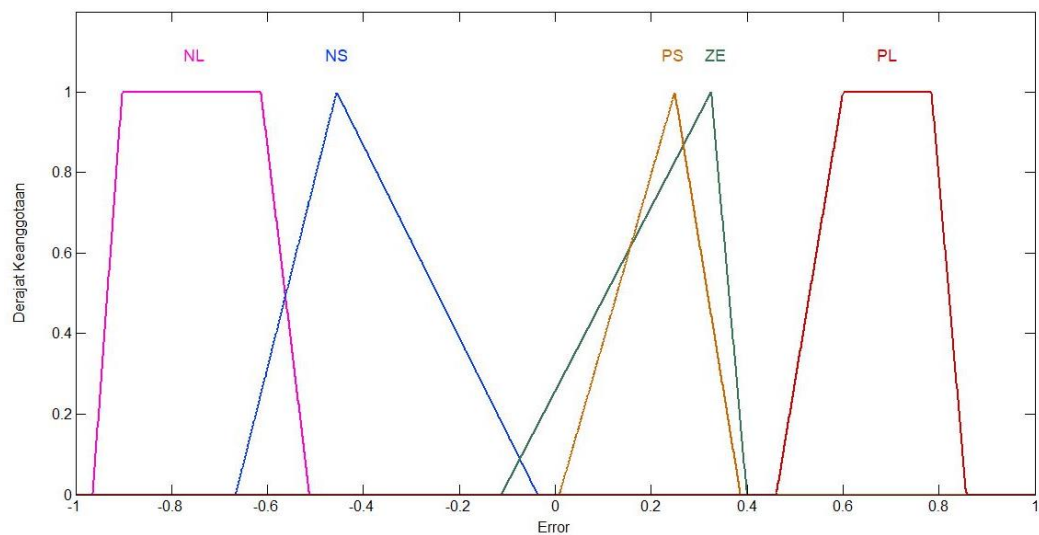
Tabel 3.11 Hasil Optimasi Parameter Fuzzy pada Variabel *Output* Sinyal Kontrol

Skenario	Parameter Fuzzy Hasil Optimasi				
	NL	NS	ZE	PS	PL
Skenario 1	[4.4297; 7.1370; 7.2205; 9.4330]	[9.4551; 9.8284; 11.0671]	[10.6299; 12.0929; 13.1362]	[12.3667; 13.5781; 14.9137]	[15.8841; 16.6588; 17.5944; 18.1680]
Skenario 2	[4.8557; 5.1173; 7.4954; 9.9047]	[8.5610; 11.1915; 11.9654]	[10.9555; 11.5876; 13.5834]	[12.1586; 13.7178; 14.6980]	[14.5526; 17.8516; 18.5052; 19.2832]
Skenario 3	[5.8671; 6.0995; 7.1876; 9.2363]	[9.3614; 9.9306; 11.0737]	[10.6711; 10.8009; 11.1066]	[12.5704; 13.8723; 14.8765]	[15.1801; 16.5820; 18.3735; 19.6261]
Skenario 4	[6.4046; 8.5382; 9.4731; 9.7540]	[8.8893; 10.7291; 10.9356]	[10.8913; 11.5414; 13.7665]	[12.0644; 13.1084; 14.7933]	[15.8869; 17.2066; 18.6474; 19.5644]

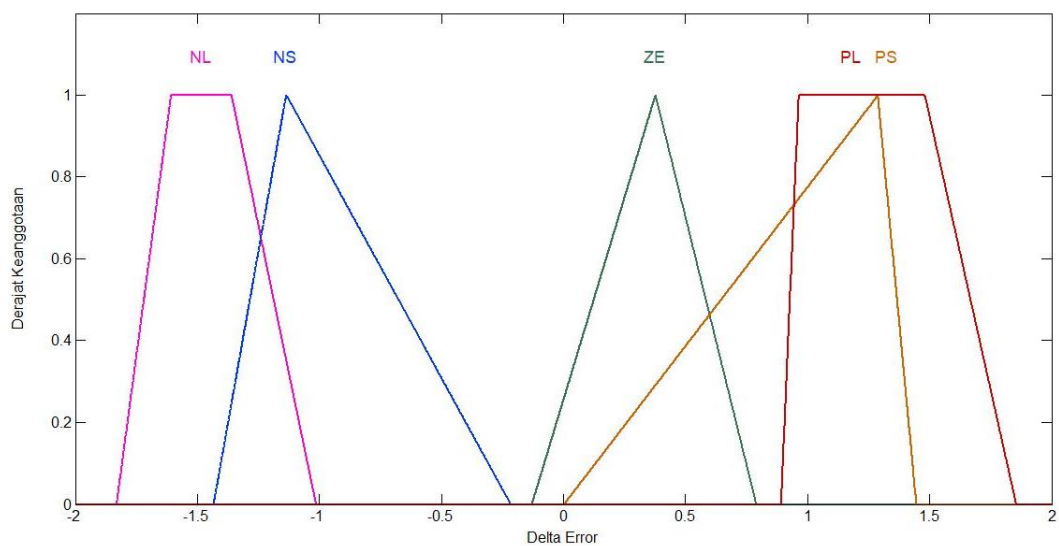
Tabel 3.11 merupakan tabel hasil optimasi parameter fuzzy dengan menggunakan *whale optimization algorithm* pada variabel fuzzy *output* sinyal kontrol. Rentang nilai *output* sinyal kontrol yang didapat dari hasil optimasi

berturut-turut dari skenario 1, skenario 2, skenario 3, dan skenario 4 adalah (4.4297, 18.1680), (4.8557, 19.2832), (5.8671, 19.6261), dan (6.4046, 19.5644).

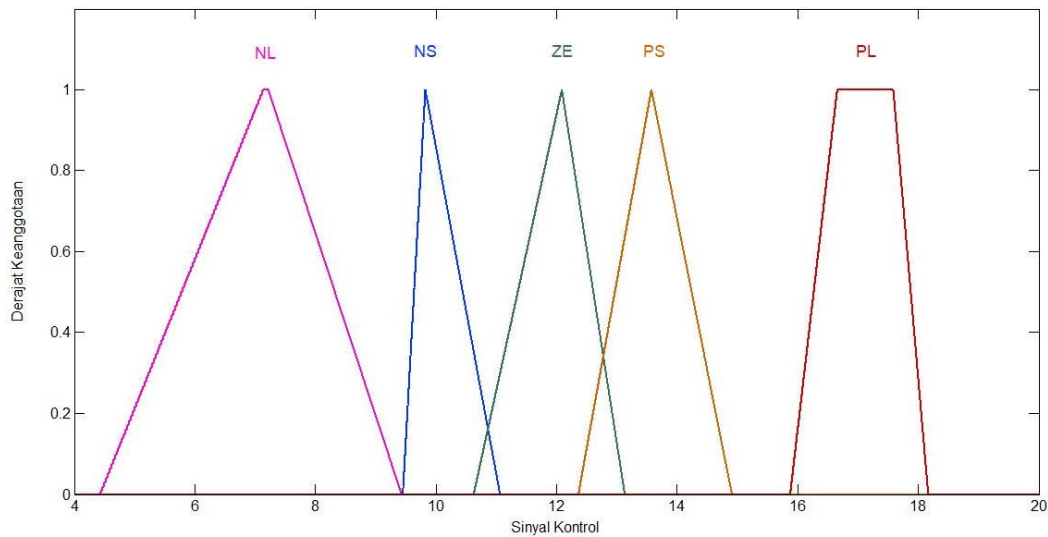
Parameter yang telah didapat dari hasil optimasi dengan menggunakan WOA, kemudian dirancang pada *toolbox* logika fuzzy di *software* Matlab. Setelah dilakukan perancangan, didapatlah bentuk fungsi keanggotaan dengan parameter fuzzy baru hasil optimasi dengan menggunakan WOA. Fungsi keanggotaan dengan parameter fuzzy yang didapat dari hasil optimasi setelah dirancang pada *toolbox* logika fuzzy di *software* Matlab ditunjukkan pada **Gambar 3.16** sampai **3.18**.



Gambar 3.16 Input Error Hasil Optimasi pada Skenario 1



Gambar 3.17 Input Delta Error Hasil Optimasi pada Skenario 1

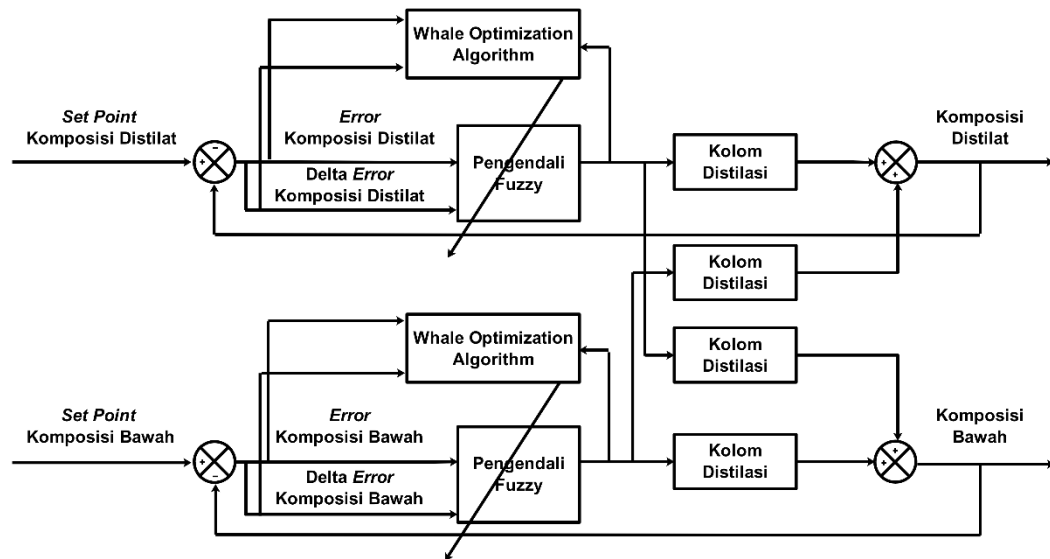


Gambar 3.18 Output Sinyal Kontrol Hasil Optimasi pada Skenario 1

Gambar 3.16, **Gambar 3.17**, dan **Gambar 3.18** secara berturut-turut adalah bentuk fungsi keanggotaan *input error*, *input delta error*, dan *output* sinyal kontrol hasil optimasi pada skenario 1. Untuk bentuk fungsi keanggotaan hasil optimasi pada skenario lainnya tersedia pada lampiran.

Dilakukan perancangan sistem pengendalian komposisi pada kolom distilasi dengan menggunakan fuzzy-WOA setelah hasil optimasi parameter pengendali fuzzy (pengendali fuzzy tanpa optimasi pada skenario 1) didapatkan. Perancangan sistem pengendalian komposisi pada kolom distilasi dengan menggunakan pengendali fuzzy-WOA hampir sama dengan perancangan sistem pengendalian komposisi pada kolom distilasi dengan menggunakan pengendali fuzzy tanpa optimasi. Perbedaannya hanya terletak pada parameter fuzzy yang digunakan pada sistem pengendalian komposisi kolom distilasi. Parameter fuzzy yang digunakan tentunya adalah parameter fuzzy hasil optimasi seperti yang ditunjukkan pada **Tabel 3.9**, **Tabel 3.10**, dan **Tabel 3.11**. Untuk *rule base* yang digunakan pada sistem pengendalian komposisi pada kolom distilasi dengan menggunakan fuzzy-WOA sama dengan *rule base* sistem pengendalian komposisi pada kolom distilasi dengan menggunakan fuzzy tanpa optimasi yang ditunjukkan pada **Tabel 3.7**. Perancangan sistem pengendalian komposisi pada kolom distilasi biner methanol-

air dengan menggunakan pengendali fuzzy-WOA ditunjukkan melalui diagram blok pada **Gambar 3.19** di bawah ini:



Gambar 3.19 Diagram Blok Sistem Pengendalian Komposisi pada Kolom Distilasi dengan Pengendali Fuzzy-WOA

Hasil perancangan disimulasi menggunakan fitur *simulink* pada Matlab setelah perancangan sistem pengendalian komposisi pada kolom distilasi dengan menggunakan pengendali fuzzy-WOA telah dilakukan. Model simulasi dari sistem pengendalian komposisi pada kolom distilasi dengan menggunakan pengendali fuzzy-WOA sama dengan model simulasi pengendali fuzzy tanpa optimasi seperti yang ditunjukkan pada **Gambar 3.14**, hanya saja simulasi dilakukan dengan mengganti *file fuzzy inference system* dengan *file fuzzy inference system* hasil optimasi pada blok pengendali fuzzy.

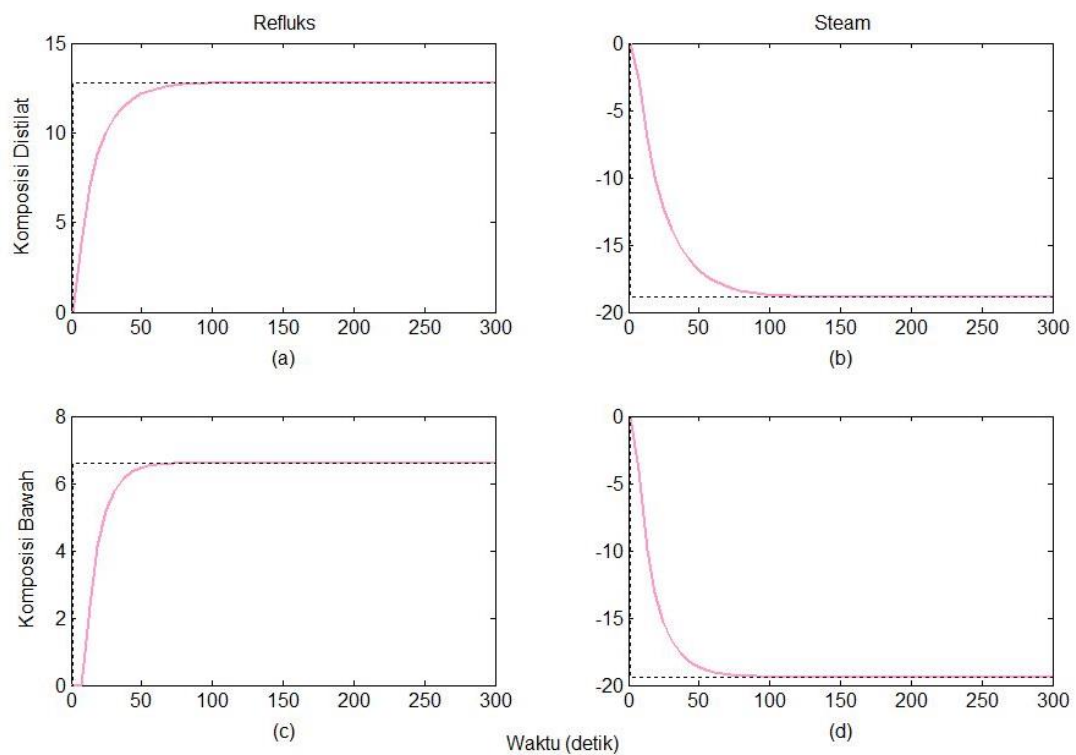
Halaman ini sengaja dikosongkan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Uji Respons *Step* Sistem *Open Loop*

Uji respons *step* sistem dalam keadaan *loop* terbuka dilakukan untuk mengetahui karakteristik dari respons *plant* disaat tidak ada *feedback* dan diberikan *input* berupa sinyal *step*.



Gambar 4.1 Respons *Step* Sistem *Open Loop* (a) Komposisi Distilat terhadap *Flow* Refluks (b) Komposisi Distilat terhadap *Flow Steam* (c) Komposisi Bawah terhadap *Flow* Refluks (d) Komposisi Bawah terhadap *Flow Steam*

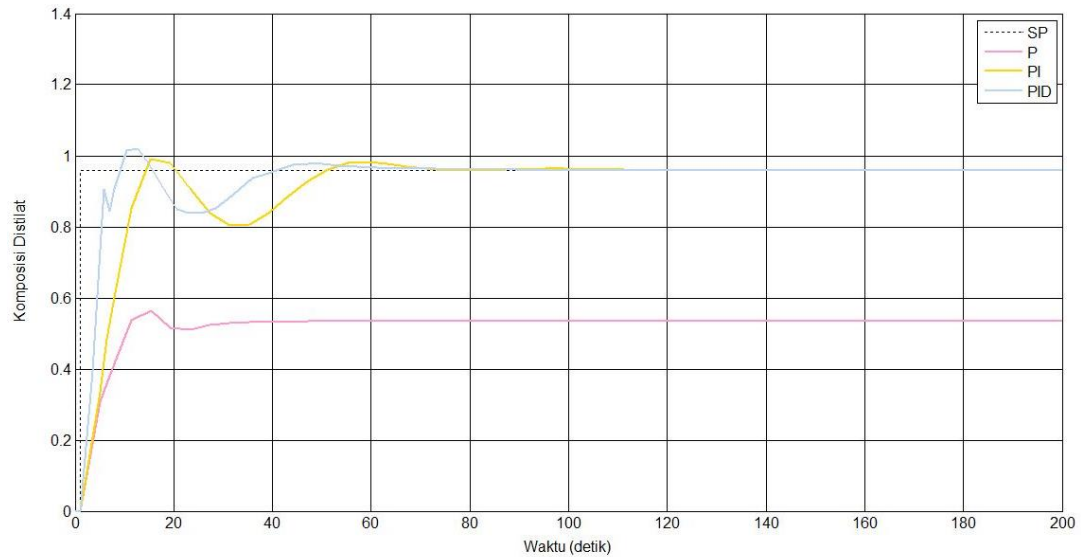
Gambar 4.1 (a) merupakan respons *step* hasil uji *open loop* komposisi distilat terhadap laju aliran refluks dan **Gambar 4.1 (b)** merupakan respons *step* hasil uji *open loop* komposisi distilat terhadap laju aliran *steam*, sedangkan

Gambar 4.1 (c) merupakan respons *step* hasil uji *open loop* komposisi bawah terhadap laju aliran refluks dan **Gambar 4.1 (d)** merupakan respons *step* hasil uji *open loop* komposisi bawah terhadap laju aliran *steam*. Pada uji respons *step* sistem *open loop*, nilai optimum dari respons yang dihasilkan akan sebesar nilai *gain* sistem/*plant* apabila *final value* sinyal *step* yang diberikan sebesar 1. Dari **Gambar 4.1** terlihat bahwasanya nilai optimum pada keempat grafik menunjukkan nilai *gain* yang sesuai dengan nilai *gain* dari masing-masing fungsi transfer kolom distilasi seperti yang ditunjukkan pada persamaan (3.6). Hal ini mengapa **Gambar 4.1 (b)** dan **Gambar 4.1 (d)** menunjukkan nilai optimum negatif karena nilai *gain* fungsi transfer **Gambar 4.1 (b)** dan **Gambar 4.1 (d)** bernilai negatif seperti yang ditunjukkan pada persamaan (3.3) dan (3.5). *Gain* negatif ini merepresentasikan karakteristik sistem yang apabila diberikan *input* berupa sinyal *step* positif akan menghasilkan *output* yang memiliki arah berlawanan. Artinya, pada keadaan *real plant*, perubahan nilai laju aliran *steam* akan menyebabkan perubahan yang berkebalikan dengan kedua nilai komposisi yang dihasilkan. Untuk respons yang tercepat dari keempat grafik dimiliki oleh **Gambar 4.1 (c)** dengan nilai *settling time* sebesar 51,27 detik. Hal ini dikarenakan **Gambar 4.1 (c)** memiliki model matematis (fungsi transfer) dengan nilai *time constant* yang paling kecil, yaitu sebesar 10,9 detik seperti yang ditunjukkan pada persamaan (3.4). Sehingga, respons dari persamaan (3.4) akan terlebih dahulu mencapai keadaan 63% daripada respons persamaan (3.2), respons persamaan (3.3), dan respons persamaan (3.5). Baik dari segi *gain* dan segi waktu keempat grafik pada **Gambar 4.1**, keduanya akan mempengaruhi karakteristik respons dari sistem pengendalian komposisi pada kolom distilasi.

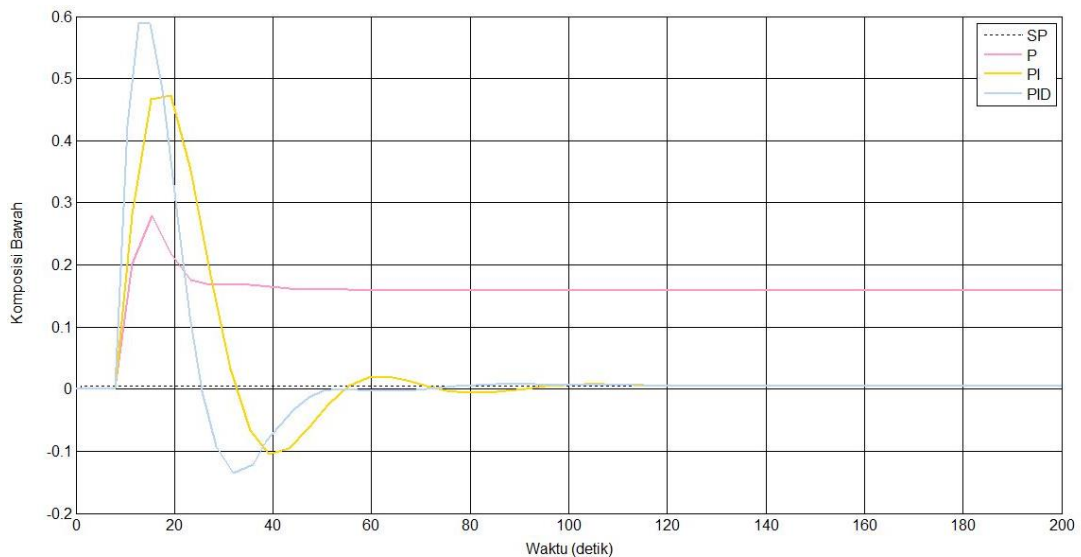
4.2 Hasil Simulasi Pengendali PID

Simulasi sistem pengendalian komposisi berbasis PID dilakukan dengan menggunakan fitur *simulink* pada *software* Matlab seperti yang ditunjukkan pada **Gambar 3.6**. Dari hasil simulasi didapat enam buah grafik respons dinamik sistem, yaitu tiga grafik respons sistem pada komposisi distilat dan tiga grafik respons sistem pada komposisi bawah. Tiga grafik respons pada komposisi distilat dengan pengendali tipe-P, tipe-PI, dan tipe-PID digabungkan ke dalam satu grafik untuk

dibandingkan begitupun juga dengan tiga grafik respons pada komposisi bawah. Grafik respons pada komposisi distilat dan komposisi bawah yang didapatkan pada perancangan pengendali PID dengan metode *autotune* ditunjukkan pada **Gambar 4.2** dan **Gambar 4.3** berikut:



Gambar 4.2 Grafik Respons Pengendali PID Komposisi Distilat



Gambar 4.3 Grafik Respons Pengendali PID Komposisi Bawah

Gambar 4.2 dan **Gambar 4.3** merupakan grafik hasil simulasi sistem pengendalian komposisi dengan menggunakan pengendali konvensional yaitu PID. **Gambar 4.2** menunjukkan grafik respons sistem pengendali terhadap komposisi

distilat, sedangkan **Gambar 4.3** menunjukkan grafik respons sistem pengendali terhadap komposisi bawah. Simulasi pengendali PID dilakukan dengan menggunakan parameter PID yang telah *dituning* seperti yang ditunjukkan pada **Tabel 3.1** dan **Tabel 3.2**. Dari kedua grafik di atas, dilakukanlah perhitungan untuk mendapatkan nilai performansi dari respons dinamik sistem yang dihasilkan. Nilai performansi dari grafik respons dinamik sistem yang didapat dengan menggunakan *syntax* pada Matlab ditunjukkan pada **Tabel 4.1** dan **Tabel 4.2** berikut:

Tabel 4.1 Performansi Pengendali PID Komposisi Distilat

Pengendali	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
P	8,09	27,31	5,78%	44,36%
PI	9,63	62,35	3,23%	0,01%
PID	4,07	36,77	6,18%	0,00%

Tabel 4.2 Performansi Pengendali PID Komposisi Bawah

Pengendali	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
P	2,19	42,15	75,43%	3090%
PI	1,73	84,85	9332%	0,38%
PID	2,09	49,10	11657%	0,18%

Tabel 4.1 adalah tabel yang menunjukkan nilai performansi hasil pengendali PID pada komposisi distilat. Pada **Tabel 4.1** terlihat bahwasanya pengendali Tipe-P memiliki *settling time* paling cepat dari pengendali Tipe-PI dan Tipe-PID, yaitu sebesar 27,31 detik. Namun, pengendali Tipe-P memiliki nilai *error steady state* yang sangat tinggi dibanding pengendali Tipe-PI dan pengendali Tipe-PID, yaitu sebesar 44,36%. Untuk *rise time* paling cepat dan *error steady state* paling kecil dimiliki oleh pengendali Tipe-PID, yaitu berturut-turut sebesar 4,07 detik dan 0%. Sedangkan, *maximum overshoot* terkecil dimiliki oleh pengendali Tipe-PI, yaitu 3,23%.

Hasil performansi pengendali PID pada komposisi bawah ditunjukkan pada **Tabel 4.2**. Pada **Tabel 4.2** menunjukkan hasil yang sangat berbeda. *Rise time* tercepat dimiliki oleh pengendali Tipe-PI dengan 1,73 detik. Untuk *settling time* tercepat dimiliki oleh pengendali Tipe-P, yaitu sebesar 42,15 detik. Serupa dengan hasil performansi pengendali pada komposisi distilat, untuk nilai *error steady state* terkecil dimiliki oleh pengendali Tipe-PID, dengan 0,18%. Perbedaan yang cukup signifikan antara hasil respons pengendali PID pada komposisi distilat dan komposisi bawah terletak pada nilai *maximum overshoot*. Nilai *maximum overshoot* yang dihasilkan oleh ketiga tipe pengendali pada komposisi bawah, menunjukkan hasil yang di luar dugaan. Secara teori, suatu sistem pengendalian haruslah memiliki nilai *maximum overshoot* di bawah 25%, namun pada kenyataannya hasil yang didapat sangat jauh dari 25%. Dari ketiga tipe pengendali, nilai *maximum overshoot* terkecil dimiliki oleh pengendali Tipe-P, yaitu sebesar 75,43%.

Jika ditinjau lebih lanjut, penyebab terjadinya nilai *maximum overshoot* yang sangat tinggi ini karena perbedaan signifikan antara nilai *set point* pada komposisi distilat dan komposisi bawah. Komposisi atas memiliki *set point* sebesar 0,96 sedangkan komposisi bawah memiliki *set point* sebesar 0,005. Perbedaan *set point* antara komposisi distilat dan komposisi bawah pada suatu sistem pengendalian MIMO akan berpengaruh terhadap nilai dan karakteristik *error* sistem yang dihasilkan. Apabila dilihat dari diagram blok sistem pengendalian MIMO pada **Gambar 3.4**, nilai *error* dari variabel proses (komposisi distilat dan komposisi bawah) secara tidak langsung saling mempengaruhi satu sama lain. Sehingga, perubahan nilai *error* yang kecil pada komposisi distilat sangat mungkin untuk menyebabkan perubahan yang sangat signifikan pada nilai variabel proses komposisi bawah. Alhasil, nilai *maximum overshoot* pada respons yang didapat akan sangat tinggi. Alasan ini juga diperkuat oleh penelitian yang dilakukan oleh (Wara et al., 2016) mengenai sistem pengendalian pada kolom distilasi biner Wood & Berry dengan menggunakan metode *internal model control*. Dengan *set point* yang sama, yaitu 0,96 pada komposisi distilat dan 0,005 pada komposisi bawah juga mendapati hasil nilai *maximum overshoot* yang tinggi. Di samping itu, (Acharya et al., n.d.) melakukan penelitian mengenai sistem pengendalian pada kolom distilasi Wood & Berry menggunakan pengendali PID dengan *set point* yang

berbeda, yaitu sebesar 1,00 pada kedua komposisi. Hasil yang didapat oleh (Acharya et al., n.d.) berbeda, penelitian (Acharya et al., n.d.) menghasilkan nilai *maximum overshoot* yang kecil. Sehingga, dari penelitian yang telah dilakukan oleh (Wara et al., 2016) dan (Acharya et al., n.d.) membuktikan bahwa perbedaan *set point* pada komposisi atas dan komposisi bawah saling mempengaruhi satu sama lain.

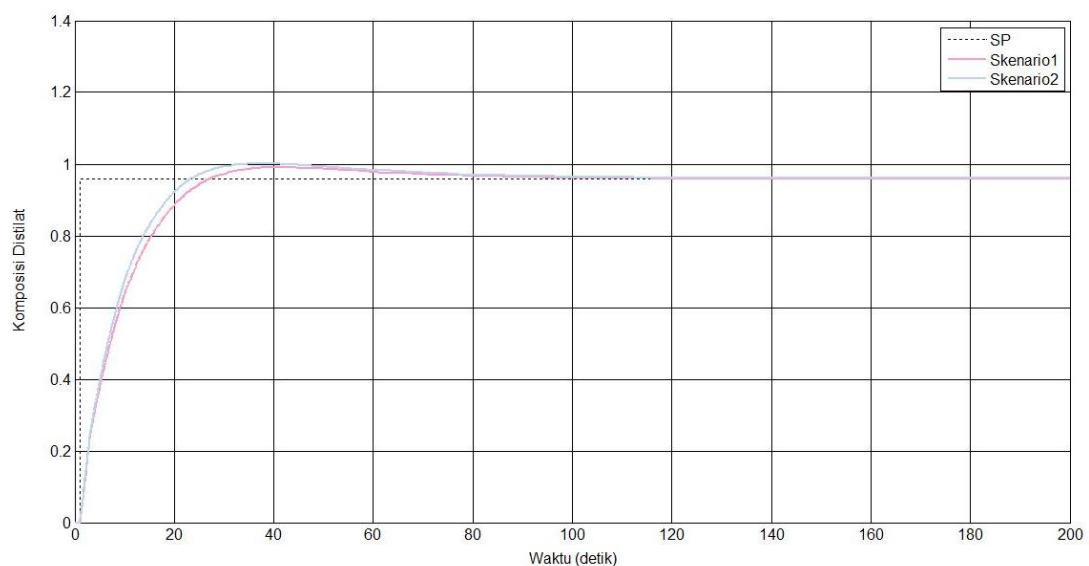
Secara garis besar, semua hasil respons dari masing-masing tipe pengendali yang didapat bersesuaian dengan karakteristik pengendali PID seperti yang telah ditunjukkan pada **Tabel 2.1**. Dari hasil yang didapat pada kedua tabel, terlihat bahwa nilai *error steady state* yang relatif kecil dimiliki oleh pengendali Tipe-PI dan pengendali Tipe-PID. Hal ini membuktikan bahwa *gain integral* pada pengendali Tipe-PI dan pengendali Tipe-PID mampu menurunkan nilai *error steady state* sistem. Sama halnya dengan *gain integral*, *gain proportional* juga berhasil membuktikan bahwa *gain proportional* mampu menurunkan *rise time* sistem.

Pada pengendalian komposisi sebuah kolom distilasi, nilai performansi dititikberatkan pada nilai *error steady state*. Hal ini dikarenakan tujuan utama dari sistem pengendalian komposisi adalah untuk meningkatkan kualitas hasil produksi (komposisi) dan menekan biaya produksi, namun terlepas dari itu semua performansi lainnya juga harus tetap dipertimbangkan. Dari **Tabel 4.1** dan **Tabel 4.2**, terlihat bahwasanya pengendali Tipe-P secara umum unggul pada dua nilai performansi, yaitu *settling time* dan *maximum overshoot*. Akan tetapi, pengendali Tipe-P memiliki *error steady state* yang sangat besar, sehingga Pengendali Tipe-P tidak dapat dikatakan sebagai tipe pengendali terbaik untuk sistem pengendalian komposisi ini. Beranjak dari pengendali Tipe-P, pengendali Tipe-PI dan pengendali Tipe-PID memiliki keunggulan yang hampir sama. Nilai *error steady state* terkecil sudah jelas dimiliki oleh pengendali Tipe-PID, namun untuk nilai *rise time* keduanya memiliki nilai yangimbang. Pengendali Tipe-PI memiliki keunggulan nilai *rise time* pada komposisi bawah dan pengendali Tipe-PID memiliki keunggulan *rise time* pada komposisi distilat. Untuk menentukan tipe pengendali terbaik, nilai *settling time* dan *maximum overshoot* antara pengendali Tipe-PI dan pengendali Tipe-PID dibandingkan. Berdasarkan teori yang dikutip dari jurnal

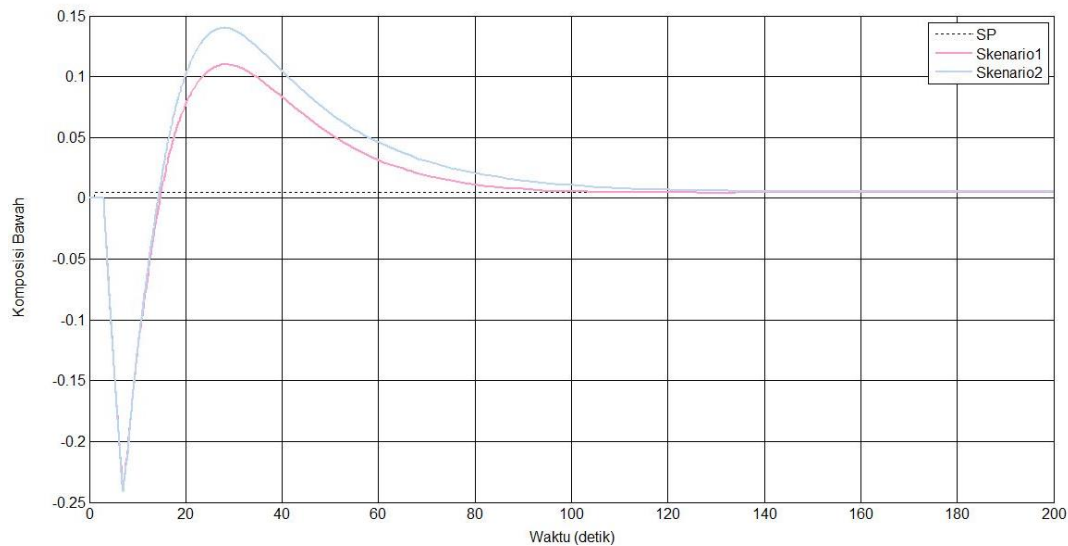
(Radita, 2017), nilai *maximum overshoot* akan berbanding terbalik dengan *settling time*. Apabila sebuah sistem memiliki *maximum overshoot* yang relatif tinggi, maka sistem tersebut akan mempunyai *settling time* yang relatif cepat. Jika ditinjau kembali, suatu sistem pengendalian mengharapkan sebuah pengendali yang dapat menghasilkan respons cepat dan juga memiliki *error steady state* kecil. Suatu sistem dengan *maximum overshoot* yang tinggi dengan nilai *settling time* yang kecil (respons cepat) akan lebih baik daripada sistem yang memiliki *maximum overshoot* lebih rendah namun dengan nilai *settling time* yang tinggi (respons lama). Sehingga, tipe pengendali PID terbaik pada sistem pengendalian komposisi kolom distilasi biner methanol-air dimiliki oleh pengendali Tipe-PID.

4.3 Hasil Simulasi Pengendali Fuzzy

Sistem pengendalian komposisi distilat dan komposisi bawah dengan pengendali fuzzy terdiri dari dua skenario seperti yang telah ditunjukkan pada **Tabel 3.4**. Kemudian, dilakukanlah simulasi sistem pengendalian berbasis logika fuzzy untuk mengetahui performansinya. Didapat dua grafik respons dari masing-masing skenario, yaitu grafik respons dari komposisi distilat dan grafik respons dari komposisi bawah. Grafik respons dari sistem pengendalian komposisi berbasis logika fuzzy ditunjukkan pada **Gambar 4.4** dan **Gambar 4.5** berikut:



Gambar 4.4 Grafik Respons Pengendali Fuzzy Komposisi Distilat



Gambar 4.5 Grafik Respons Pengendali Fuzzy Komposisi Bawah

Gambar 4.5 dan **Gambar 4.6** merupakan grafik hasil simulasi sistem pengendalian komposisi dengan menggunakan sistem logika fuzzy sebagai pengendalinya. **Gambar 4.5** merupakan grafik respons sistem pengendali fuzzy terhadap komposisi distilat, sedangkan **Gambar 4.6** merupakan grafik respons sistem pengendali fuzzy terhadap komposisi bawah. Jika diamati lebih lanjut, pada **Gambar 4.6** terdapat adanya *inverse response*. *Inverse response* terjadi ketika terdapat dua sistem atau proses berbeda terjadi secara bersamaan, namun dengan dinamika dan arah yang berlawanan. Persamaan (3.4) memiliki arah yang berbeda dengan persamaan (3.5). Ketika respons yang dihasilkan oleh persamaan (3.4) dan (3.5) dijumlahkan seperti ditunjukkan pada **Gambar 3.12**, maka hasil akhir respons yang didapat akan mengikuti respons dari persamaan yang memiliki amplitudo lebih besar. Untuk memvalidasi hasil respons yang didapat adalah benar, dilakukan perbandingan hasil respons yang didapat dengan hasil respons penelitian lain. (Singh et al., 2011) melakukan penelitian mengenai sistem pengendalian komposisi pada kolom distilasi dengan menggunakan logika fuzzy sebagai pengendalinya. Dengan *set point* yang sama serta jumlah himpunan fuzzy dan *rule base* yang berbeda, karakteristik respons yang dihasilkan oleh penelitian (Singh et al., 2011) menunjukkan kesamaan yaitu terdapat adanya *inverse response* pada hasil komposisi bawahnya. Kemudian, untuk menentukan skenario mana yang

menghasilkan respons terbaik, dilakukanlah perhitungan nilai performansi dari respons yang dihasilkan oleh masing-masing skenario. Nilai performansi dari grafik respons dinamik sistem yang didapat dengan menggunakan *syntax* pada Matlab ditunjukkan pada **Tabel 4.3** dan **Tabel 4.4** berikut:

Tabel 4.3 Performansi Pengendali Fuzzy Komposisi Distilat

Skenario	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
Skenario 1	16,68	59,82	3,39%	0,13%
Skenario 2	14,71	65,98	4,29%	0,00%

Tabel 4.4 Performansi Pengendali Fuzzy Komposisi Bawah

Skenario	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
Skenario 1	0,19	83,27	2089%	0,54%
Skenario 2	0,19	100,34	2353%	14,38%

Tabel 4.3 dan **Tabel 4.4** merupakan tabel yang berisi hasil perhitungan nilai performansi dari sistem pengendalian komposisi berbasis logika fuzzy. Pada **Tabel 4.3**, diketahui bahwa skenario 1 memiliki *settling time* yang lebih cepat dan *maximum overshoot* yang lebih kecil dibandingkan skenario 2. Namun, nilai *settling time* dan *maximum overshoot* pada kedua skenario tidak menunjukkan perbedaan yang signifikan, yaitu berturut-turut sebesar 59,82 detik dan 3,39% pada skenario 1 serta 65,98 detik dan 4,29% pada skenario 2. Disisi lain, skenario 2 memiliki *rise rime* yang lebih cepat dan *error steady state* yang lebih kecil dibandingkan skenario 1. Namun juga, nilai *rise time* dan *error steady state* pada kedua skenario tidak menunjukkan perbedaan yang signifikan, yaitu berturut-turut sebesar 16,68 detik dan 0,13% pada skenario 1 serta 14,71 detik dan 0,00% pada skenario 2.

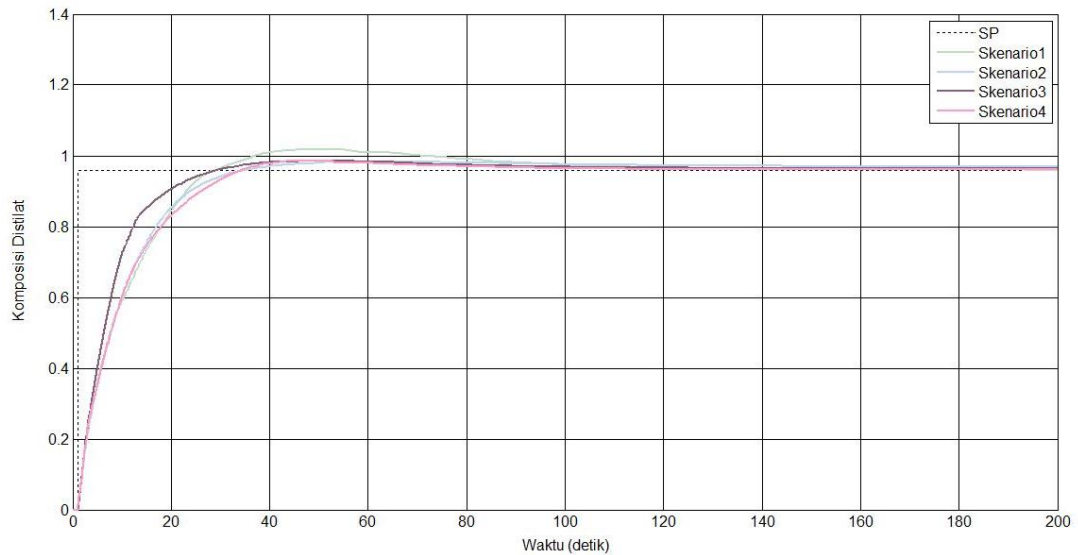
Tabel 4.4 menunjukkan hasil yang sedikit berbeda. Jika pada **Tabel 4.3** *rise time* tercepat dan nilai *error steady state* terkecil dimiliki oleh skenario 2, pada **Tabel 4.4** nilai *rise time* pada kedua skenario menunjukkan kesamaan yaitu 0,19 detik. Sedangkan, nilai *error steady state* terkecil dimiliki oleh skenario 1, yaitu sebesar 0,54%. Untuk nilai *maximum overshoot* terkecil tetap dimiliki oleh skenario 1, yaitu sebesar 2089%. Jika diamati, besar nilai *maximum overshoot* pada kedua skenario juga serupa dengan besar nilai *maximum overshoot* pengendali PID pada **Tabel 4.2**. Nilai *maximum overshoot* yang didapat dari hasil pengendali fuzzy pada komposisi bawah jauh dari 25%. Namun jika dibandingkan dengan nilai *maximum overshoot* milik pengendali PID, nilai *maximum overshoot* milik pengendali fuzzy lebih kecil. Untuk *settling time* tercepat juga tetap dimiliki oleh skenario 1, yaitu sebesar 83,27 detik.

Dari kedua tabel tersebut, dilakukan perbandingan terhadap nilai performansi yang dihasilkan oleh kedua skenario. Pada **Tabel 4.3**, skenario 1 memiliki keunggulan nilai *maximum overshoot* yang lebih kecil dan *settling time* yang lebih cepat dibandingkan skenario 2. Sedangkan skenario 2, memiliki keunggulan nilai *error steady state* yang lebih kecil dan *rise time* yang lebih cepat dibandingkan skenario 1. Namun, nilai performansi dari keduanya tidak menunjukkan perbedaan yang signifikan. Pada **Tabel 4.4**, terlihat jelas bahwasanya skenario 1 lebih unggul dari pada skenario 2. Skenario 1 memiliki *settling time* yang lebih cepat serta *error steady state* dan *maximum overshoot* yang lebih kecil dibandingkan skenario 2. Sehingga, disimpulkan bahwasanya pengendali fuzzy pada skenario 1 lah yang memiliki nilai performansi lebih baik.

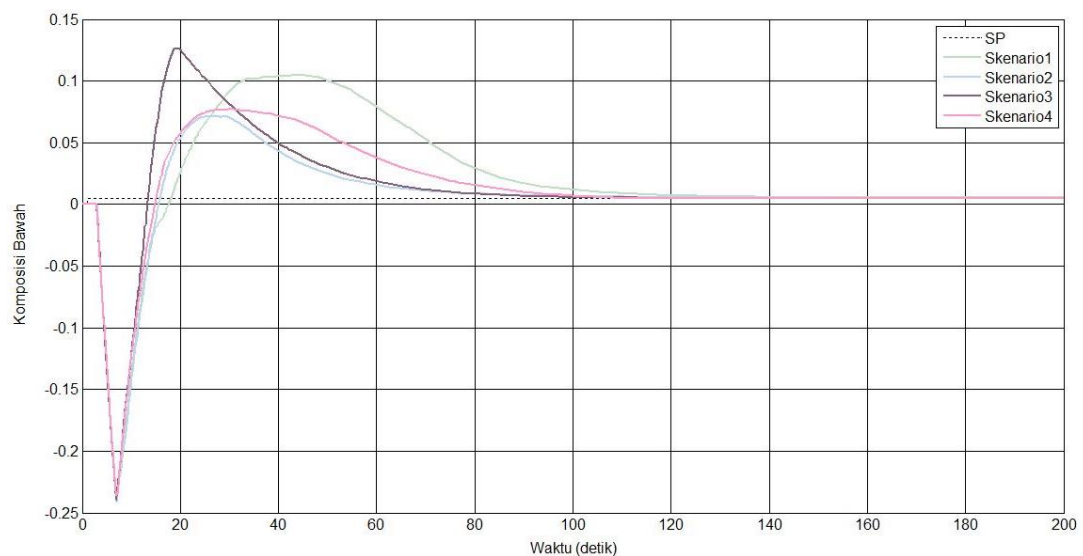
4.4 Hasil Simulasi Pengendali Fuzzy-WOA

Sistem pengendalian komposisi distilat dan komposisi bawah dengan pengendali fuzzy-WOA terdiri dari empat skenario seperti yang telah ditunjukkan pada **Tabel 3.8**. Setelah dilakukan proses optimasi menggunakan WOA pada parameter pengendali fuzzy skenario terbaik (pengendali fuzzy tanpa optimasi dengan hasil performansi terbaik), dilakukanlah simulasi sistem pengendalian berbasis logika fuzzy-WOA untuk mengetahui performansinya. Didapat dua grafik respons dari masing-masing skenario, yaitu grafik respons dari komposisi distilat

dan grafik respons dari komposisi bawah. Grafik respons dari sistem pengendalian komposisi berbasis logika fuzzy-WOA ditunjukkan pada **Gambar 4.6** dan **Gambar 4.7** berikut:



Gambar 4.6 Grafik Respons Pengendali Fuzzy-WOA Komposisi Distilat



Gambar 4.7 Grafik Respons Pengendali Fuzzy-WOA Komposisi Bawah

Gambar 4.6 dan **Gambar 4.7** merupakan grafik hasil simulasi sistem pengendalian komposisi dengan menggunakan pengendali fuzzy-WOA. **Gambar 4.6** menunjukkan grafik respons sistem pengendali terhadap komposisi distilat,

sedangkan **Gambar 4.7** menunjukkan grafik respons sistem pengendali terhadap komposisi bawah. Simulasi pengendali fuzzy-WOA dilakukan dengan menggunakan parameter pengendali fuzzy dengan performansi terbaik yang telah dioptimasi seperti yang ditunjukkan pada **Tabel 3.9**, **Tabel 3.10**, dan **Tabel 3.11**. Sama halnya dengan grafik pada **Gambar 4.5**, grafik pada **Gambar 4.6** juga memiliki *inverse response* pada semua skenario. Seperti yang dijelaskan sebelumnya, bahwasanya *inverse response* terjadi ketika terdapat dua sistem atau proses berbeda terjadi secara bersamaan, namun dengan dinamika dan arah yang berlawanan. Untuk menentukan skenario terbaik pada pengendali fuzzy-WOA, dilakukanlah perhitungan nilai performansi dari grafik respons yang dihasilkan oleh masing-masing skenario. Nilai performansi dari grafik respons dinamik sistem ditunjukkan pada **Tabel 4.5** dan **Tabel 4.6**:

Tabel 4.5 Performansi Pengendali Fuzzy-WOA Komposisi Distilat

Skenario	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
Skenario 1	19,17	93,41	6,21%	0,00%
Skenario 2	19,47	31,86	1,55%	0,98%
Skenario 3	14,30	58,30	2,36%	0,40%
Skenario 4	20,80	62,05	2,80%	0,14%

Tabel 4.6 Performansi Pengendali Fuzzy-WOA Komposisi Bawah

Skenario	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
Skenario 1	0,38	105,93	1830%	8,78%
Skenario 2	0,22	74,30	1328%	0,30%
Skenario 3	0,11	76,44	2423%	0,00%
Skenario 4	0,21	91,34	1443%	0,00%

Tabel 4.5 adalah tabel yang memuat nilai performansi pengendali fuzzy-WOA pada komposisi distilat. Pada **Tabel 4.5**, terlihat bahwasanya *rise time* tercepat dimiliki oleh skenario 3 yaitu sebesar 14,30 detik. Untuk nilai *error steady state* terkecil dimiliki oleh skenario 1 yaitu sebesar 0,00%. Sedangkan, untuk

settling time tercepat dan nilai *maximum overshoot* terkecil dimiliki oleh skenario 2 yaitu berturut-turut sebesar 31,86 detik dan 1,55%. Jika diperhatikan, sebenarnya nilai hasil performansi pengendali fuzzy-WOA dari keempat skenario tidak menunjukkan perbedaan yang signifikan.

Nilai hasil performansi pengendali fuzzy-WOA pada komposisi bawah ditunjukkan pada **Tabel 4.6**. Serupa dengan **Tabel 4.5**, *rise time* tercepat dimiliki oleh skenario 3 yaitu sebesar 0,11 detik. Namun, nilai *error steady state* terkecil pada komposisi bawah dimiliki oleh skenario 3 dan skenario 4 yaitu sebesar 0,00%. Sama halnya dengan *rise time*, untuk *settling time* tercepat dan nilai *maximum overshoot* terkecil serupa dengan **Tabel 4.5** juga tetap dimiliki oleh skenario 2 yaitu berturut-turut sebesar 74,30 detik dan 1328%.

Jika **Tabel 4.6** diperhatikan, untuk nilai *maximum overshoot* keempat skenario yang didapat oleh pengendali fuzzy-WOA masih tetap menunjukkan nilai yang jauh dari 25% seperti milik pengendali PID dan pengendali fuzzy tanpa optimasi yang berturut-turut ditunjukkan pada **Tabel 4.2** dan **Tabel 4.4**. Namun jika dibandingkan dengan nilai *maximum overshoot* milik pengendali PID, nilai *maximum overshoot* milik pengendali fuzzy-WOA menunjukkan penurunan dan ternilai jauh lebih baik. Penurunan nilai *maximum overshoot* ini berdampak pada nilai *settling time* sistem. Seperti yang telah dijelaskan pada **Sub BAB 4.2**, teori yang dikutip dari jurnal milik (Radita, 2017) mengatakan bahwa nilai *maximum overshoot* akan selalu berkebalikan dengan *settling time*. Jika *settling time* dan nilai *maximum overshoot* milik pengendali PID dan fuzzy-WOA dibandingkan, terbukti bahwa dengan nilai *maximum overshoot* yang jauh lebih kecil akan menghasilkan *settling time* yang relatif lebih lama. Berlaku juga sebaliknya, respons dengan nilai *maximum overshoot* yang jauh lebih besar akan menghasilkan *settling time* yang relatif lebih cepat.

Dari kedua tabel tersebut, dilakukan perbandingan terhadap nilai performansi yang dihasilkan oleh keempat skenario. Pada **Tabel 4.5**, skenario 1 memiliki keunggulan nilai *error steady state* yang lebih kecil dibandingkan ketiga skenario lainnya. Sedangkan skenario 2 lebih unggul pada nilai *maximum overshoot* dan *settling time* dibandingkan ketiga skenario lainnya. Untuk keunggulan pada nilai *rise time* dimiliki oleh skenario 3. Pada **Tabel 4.6**, skenario 2 masih tetap unggul

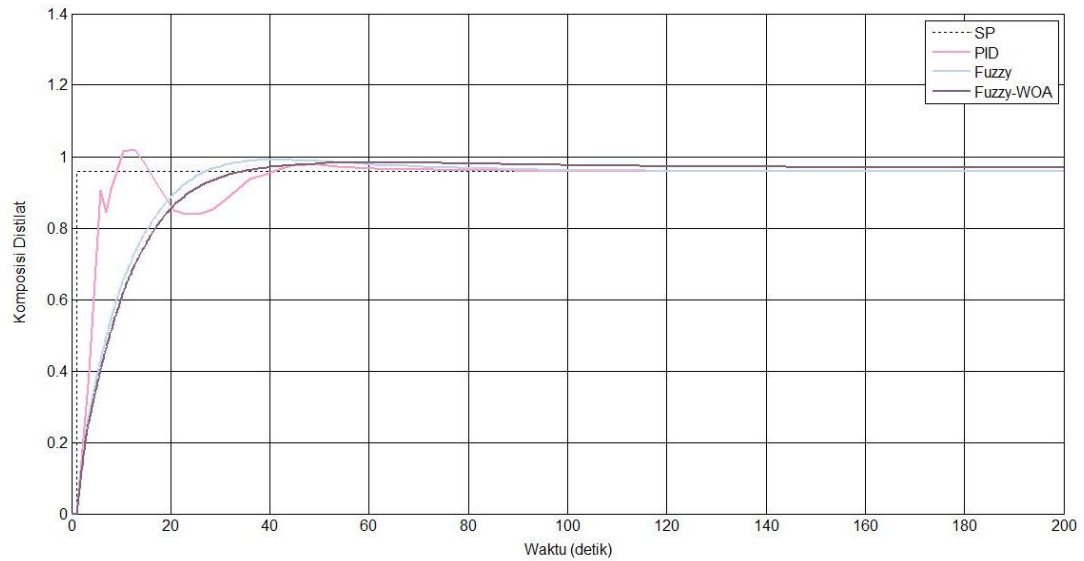
pada nilai *maximum overshoot* dan *settling time* dibandingkan ketiga skenario lainnya. Untuk keunggulan pada nilai *rise time* juga tetap dimiliki oleh skenario 3. Namun berbeda dengan **Tabel 4.5**, pada **Tabel 4.6** keunggulan pada nilai *error steady state* dimiliki oleh skenario 3 dan skenario 4. Dari performansi yang dihasilkan setiap skenario pada kedua tabel, terlihat bahwasanya skenario 2 memiliki konsistensi hasil yang baik terutama pada nilai *maximum overshoot* dan *settling time*. Jika ditinjau dari nilai *error steady state* dan *rise time* yang dihasilkan pun, skenario 2 memiliki nilai *error steady state* dan *rise time* yang cukup bagus dan tidak menunjukkan perbedaan yang signifikan jika dibandingkan dengan skenario lainnya. Disimpulkan bahwasanya pengendali fuzzy-WOA terbaik dimiliki oleh skenario 2.

4.5 Perbandingan Hasil Simulasi Ketiga Metode Pengendali

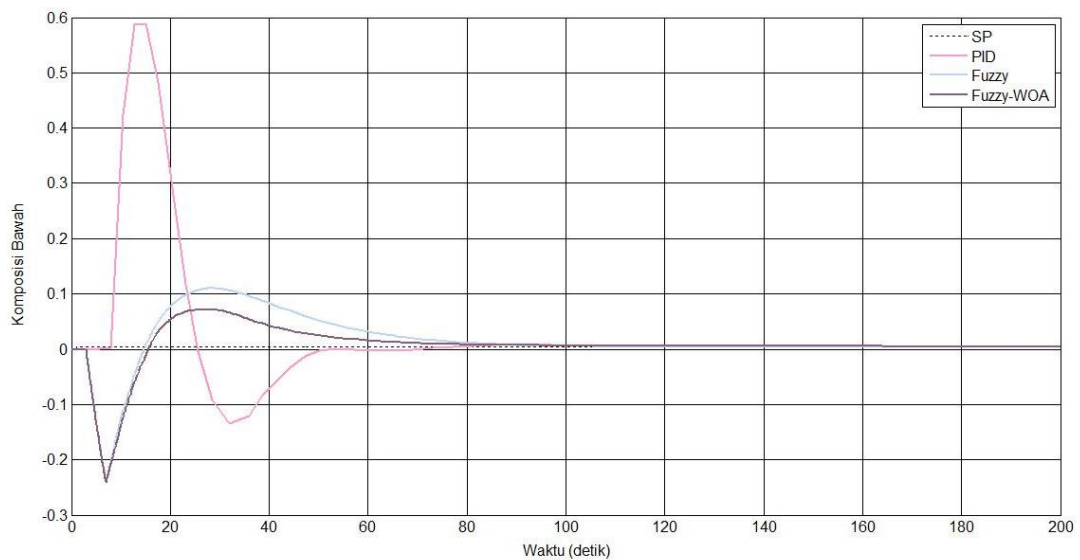
Perancangan sistem pengendali pada penelitian ini terdiri dari tiga metode, yaitu sistem pengendali konvensional PID, sistem pengendali dengan sistem logika fuzzy tanpa optimasi, dan sistem pengendali dengan sistem logika fuzzy-WOA. Setiap metode pengendali memiliki sejumlah tipe atau skenario tertentu, dimana nilai hasil performansi dari tipe atau skenario tersebut dibandingkan untuk mengetahui tipe atau skenario mana yang menunjukkan nilai hasil performansi terbaik. Pada tugas akhir ini, pengendali konvensional PID memiliki tiga rancang tipe pengendali, yaitu tipe-P, tipe-PI, dan tipe-PID. Dari ketiga tipe pengendali PID yang digunakan, performansi terbaik dimiliki oleh pengendali tipe-PID. Untuk pengendali fuzzy tanpa optimasi memiliki dua buah skenario rancangan, seperti yang ditunjukkan pada **Tabel 3.6**. Pada pengendali fuzzy tanpa optimasi, performansi terbaik dimiliki oleh skenario 1. Sedangkan, untuk pengendali fuzzy-WOA memiliki empat skenario rancangan seperti yang ditunjukkan pada **Tabel 3.8**. Untuk pengendali fuzzy-WOA, performansi terbaik dimiliki oleh skenario 2.

Nilai hasil performansi terbaik dari masing-masing metode pengendali dibandingkan satu sama lain, hal ini dilakukan untuk menentukan metode pengendali mana yang menghasilkan performansi pengendali terbaik. Untuk melihat perbedaannya, grafik dari ketiga metode pengendali digabungkan. Grafik perbandingan dari ketiga metode pengendali dengan tipe dan skenario terbaiknya,

yaitu pengendali Tipe-PID, pengendali fuzzy tanpa optimasi, dan pengendali fuzzy-WOA ditunjukkan pada **Gambar 4.8** dan **Gambar 4.9** berikut:



Gambar 4.8 Grafik Respons Komposisi Distilat Ketiga Metode



Gambar 4.9 Grafik Respons Komposisi Bawah Ketiga Metode

Gambar 4.8 menunjukkan grafik respons dari ketiga metode pengendali terhadap komposisi distilat, sedangkan **Gambar 4.9** menunjukkan grafik respons dari ketiga metode pengendali terhadap komposisi bawah. Pada kedua gambar, terlihat bahwasanya pengendali PID menunjukkan adanya osilasi pada grafik responsnya. Untuk mengetahui lebih jelas metode pengendali mana yang terbaik,

dilakukan perbandingan terhadap nilai performansi yang dihasilkan. Perbandingan nilai performansi dari ketiga metode pengendali dengan tipe dan skenario terbaiknya ditunjukkan pada **Tabel 4.7** dan **Tabel 4.8** berikut:

Tabel 4.7 Performansi Ketiga Metode Pengendali Komposisi Distilat Terbaik

Pengendali	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
PID	4,07	36,77	6,18%	0,00%
Fuzzy	16,68	59,82	3,39%	0,13%
Fuzzy-WOA	19,47	31,86	1,55%	0,98%

Tabel 4.8 Performansi Ketiga Metode Pengendali Komposisi Bawah Terbaik

Pengendali	<i>Rise Time</i> (detik)	<i>Settling Time</i> (detik)	<i>Maximum Overshoot</i>	<i>Error Steady State</i>
PID	2,09	49,10	11657%	0,18%
Fuzzy	0,19	83,27	2089%	0,54%
Fuzzy-WOA	0,22	74,30	1328%	0,30%

Tabel 4.7 menunjukkan perbandingan nilai performansi ketiga metode pengendali pada komposisi distilat. Pada **Tabel 4.7** terlihat bahwasanya *rise time* tercepat dan nilai *error steady state* terkecil dimiliki oleh pengendali PID, yaitu berturut-turut sebesar 4,07 detik dan 0,00%. Untuk *settling time* tercepat dan nilai *maximum overshoot* terkecil dimiliki oleh pengendali fuzzy-WOA, yaitu berturut-turut sebesar 31,86 detik dan 1,55%. Untuk perbandingan nilai performansi hasil ketiga metode pengendali pada komposisi bawah ditunjukkan oleh **Tabel 4.8**. Pada **Tabel 4.8** terdapat beberapa perbedaan untuk *rise time* tercepat dimiliki oleh pengendali fuzzy tanpa optimasi yaitu sebesar 0,19 detik. Sedangkan, untuk *settling time* tercepat dimiliki oleh pengendali PID. Namun, untuk nilai *error steady state* dan nilai *maximum overshoot* terkecil berturut-turut tetap dimiliki oleh pengendali PID dan pengendali fuzzy-WOA, yaitu sebesar 0,18% dan 1328%.

Dari kedua tabel, terlihat bahwasanya pengendali PID memiliki keunggulan pada nilai *error steady state* baik pada komposisi distilat maupun pada komposisi

bawah. Namun jika nilai *error steady state* dari ketiga pengendali dibandingkan, nilai *error steady state* milik pengendali fuzzy tanpa optimasi dan pengendali fuzzy-WOA tidak menunjukkan perbedaan yang signifikan dengan nilai *error steady state* milik pengendali PID. Artinya jika dibandingkan dengan pengendali PID, pengendali fuzzy tanpa optimasi dan pengendali fuzzy-WOA juga mampu untuk menghasilkan nilai *error steady state* yang kecil, khususnya pada sistem pengendalian komposisi kolom distilasi biner methanol-air Wood & Berry. Untuk nilai *rise time*, keunggulan pada respons komposisi distilat dimiliki oleh pengendali PID sedangkan pada respons komposisi bawah dimiliki oleh pengendali fuzzy tanpa optimasi. Secara garis besar ketiga pengendali menghasilkan nilai *rise time* yang tidak jauh berbeda satu sama lain. Selanjutnya, keunggulan pada nilai *settling time* pada komposisi distilat dimiliki oleh pengendali fuzzy-WOA, sedangkan pada komposisi bawah dimiliki oleh pengendali PID. Untuk keunggulan nilai *maximum overshoot* dimiliki oleh pengendali fuzzy-WOA, baik pada komposisi distilat maupun komposisi bawah. Jika ditinjau lebih lanjut, secara keseluruhan perbedaan yang sangat terlihat jelas, terletak pada nilai *maximum overshoot*. Pengendali fuzzy (baik pengendali fuzzy tanpa optimasi maupun pengendali fuzzy-WOA) terlihat lebih unggul dalam menghasilkan respons dengan nilai *maximum overshoot* yang lebih kecil. Namun kinerja pengendali fuzzy dalam menghasilkan respons dengan nilai *maximum overshoot* yang kecil, berdampak pada nilai *settling time*-nya. Hal ini dikarenakan nilai *maximum overshoot* akan selalu berkebalikan dengan nilai *settling time*. Nilai *maximum overshoot* yang tinggi ini wajar terjadi dikarenakan perbedaan *set point* yang sangat besar antara komposisi distilat dan komposisi bawah. Nilai *set point* yang sangat kecil milik komposisi bawah ini, menyebabkan nilai *error* yang didapatkan oleh komposisi bawah akan terkesan/terlihat sangat besar walaupun sebenarnya perbedaan antara nilai *set point* dengan nilai variabel proses sangat kecil. Dari perbandingan hasil performansi yang didapatkan oleh ketiga pengendali, disimpulkan bahwasanya secara keseluruhan pengendali fuzzy-WOA memiliki performansi yang lebih baik dari pada pengendali PID dan pengendali fuzzy tanpa optimasi.

Halaman ini sengaja dikosongkan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapat dari hasil penelitian pada tugas akhir ini sebagai berikut:

1. Nilai parameter dari pengendali *fuzzy-whale optimization algorithm* yang menghasilkan nilai hasil performansi terbaik ditunjukkan pada **Tabel 3.9**, **Tabel 3.10**, dan **Tabel 3.11** tepatnya pada baris skenario 2 dengan parameter WOA berupa *search agent* dan iterasi maksimum berturut-turut sebesar 1000 dan 500.
2. Pengendali fuzzy-WOA mampu menghasilkan performansi yang lebih baik dibandingkan dengan pengendali PID dan pengendali fuzzy tanpa optimasi. Nilai performansi pengendali fuzzy-WOA pada komposisi distilat yang didapat yaitu dari *rise time*, *settling time*, *maximum overshoot*, dan *error steady state* berturut-turut sebesar 16,68 detik, 31,86 detik, 1,55%, dan 0,98%. Untuk nilai performansi pengendali fuzzy-WOA pada komposisi bawah yang didapat yaitu dari *rise time*, *settling time*, *maximum overshoot*, dan *error steady state* berturut-turut sebesar 0,22 detik, 74,30 detik, 1328%, dan 0,30%.

5.2 Saran

Saran yang dapat diberikan terkait hasil tugas akhir ini untuk pengembangan penelitian selanjutnya sebagai berikut:

1. Sistem pengendalian MIMO dengan *decoupling* bisa digunakan untuk menghilangkan interaksi antar variabel pada *plant* kolom distilasi.
2. Optimasi dapat diterapkan atau diaplikasikan untuk mendapatkan bentuk fungsi keanggotaan dan *rule base* sistem logika fuzzy yang optimal.
3. Menerapkan sistem logika fuzzy Tipe-2 dan *adaptive fuzzy* untuk mendapatkan hasil respons dinamik yang lebih baik dan beragam.

4. Menerapkan variasi algoritma optimasi seperti *genetic algorithm* (GA), *ant colony optimization* (ACO), *modified whale optimization algorithm* (MWOA), dan *particle swarm optimization* (PSO) untuk mengetahui algoritma mana yang mampu menghasilkan parameter fuzzy terbaik pada sistem pengendalian komposisi kolom distilasi biner methanol-air.

DAFTAR PUSTAKA

- Acharya, P., Dumpa, G., & Dan, T. K. (n.d.). *Modelling and control of distillation column. 1.*
- Agustin, A. H., Gandhiadi, G. K., & Oka, T. B. (2016). Penerapan Metode Fuzzy Sugeno Untuk Menentukan Harga Jual Sepeda Motor Bekas. *E-Jurnal Matematika*, 5(4), 176.
- Aisjah, A. S., & Arifin, S. (2018). *Sistem Pengendalian Otomatis: Perkembangan Implementasi di Era Revolusi Industri 1.0 - 4.0.*
- Biyanto, T. R., Wahyudi, H., & Santoso, H. H. (2015). *Strategi Kontrol Kolom Distilasi Tunggal Sistem Biner Metanol- Strategi Kontrol Kolom Distilasi Tunggal Sistem Biner Metanol-Air.*
- Buckley, P. S., Luyben, W. L., & Shunta, J. P. (1985). *Design of Distillation Column Control Systems.*
- Chen, D., & Seborg, D. E. (2002). Relative Gain Array Analysis for Uncertain Process Models. *AIChE Journal*, 48(2), 302–310.
- Dorrah, H. T., & El-garhy, A. M. (2012). PSO based optimized fuzzy controllers for decoupled highly interacted distillation process. *Ain Shams Engineering Journal*, 3(3), 251–266.
- Dorteus, L. R. (2015). Penerapan Logika Fuzzy Metode Sugeno Berdasarkan Data Persediaan Dan Jumlah Permintaan (Studi Kasus : Pabrik Roti Sarinda Ambon) Application of Fuzzy Logic Method Sugeno To Determine the Total Production of Bread ,. *Jurnal Ilmu Matematika Dan Terapan*, 9(2), 121–134.
- Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation*, 48(November 2018), 1–24.
- Guru, R., Arumugam, A., Balasubramanian, G., & Balaji, V. S. (2013). Fuzzy Based Composition Control of Distillation Column. *International Journal of Engineering and Technology*, 5(2), 1863–1869.
- Hakim, W., Turmudi, & Irawan, W. H. (2015). *Sistem Pendukung Keputusan Metode Sugeno dalam Menentukan Tingkat Kepribadian Siswa berdasarkan*

- Pendidikan*. 4(1).
- Halvorsen, I. J., & Skogestad, S. (2000). Chapter 2 Distillation Theory. In *Encyclopedia of Separation Science* (Issue August).
- Harefa, K. (2017). Penerapan Fuzzy Inference System Untuk Menentukan Jumlah Pembelian Produk Berdasarkan Data Persediaan dan Penjualan. *Jurnal Informatika Universitas Pamulang*, 2(4).
- Haris, M. S., Santoso, E., & Ratnawati, D. E. (2010). *Implementasi Metode Fuzzy Time Series Dengan Penentuan Interval Berbasis Rata-Rata Untuk Peramalan Data Penjualan Bulanan*. May 2019, 210.
- Hasanah, S. N., & Widiastuti, N. I. (2014). Representasi Emosi Menggunakan Logika Fuzzy Pada Permainan Bonny's Tooth Booth. *Jurnal Ilmiah Komputer Dan Informatika*, 3(2).
- Husnaini, I., & Krismadinata. (2017). Komparasi Pengendali PI dan PID untuk Tegangan Keluaran Konverter Buck. *Jurnal Nasional Teknik Elektro*, 6(3).
- Komariah, L. N., Ramdja, A. F., & Leonard, N. (2009). Tinjauan Teoritis Perancangan Kolom Distilasi Untuk Pra-Rencana Pabrik Skala Industri. *Jurnal Teknik Kimia*, 16(4), 19–27.
- Kusumadewi, S., & Purnomo, H. (2010). *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*.
- Mahmoud, M. M. A. S. (2018). Damping of Surge Effects and Enhancement of Dynamic Response of Distillation Plant Using MIMO Fuzzy Control. *IFAC-PapersOnLine*, 51(30), 15–20.
- Margi S, K., & Pendawa W, S. (2015). Analisa Dan Penerapan Metode Single Exponential Smoothing Untuk Prediksi Penjualan Pada Periode Tertentu. *Prosiding SNATIF*, 1998, 259–266.
- Morshidi, M. N., Musirin, I., Rahim, S. R. A., Adzman, M. R., & Hussain, M. H. (2018). Whale Optimization Algorithm Based Technique for Distributed Generation Installation in Distribution System. *Bulletin of Electrical Engineering and Informatics*, 7(3), 442–449.
- Muzwar, H. S. N., Pamososuryo, A. K., & Ekawati, E. (2014). *Pemodelan Kolom Distilasi Pabrik Petrokimia dengan Menggunakan Distributed Control System*. 6(2), 85–94.

- Patel, N. C., & Debnath, M. K. (2019). Whale Optimization Algorithm Tuned Fuzzy Integrated PI Controller for LFC Problem in Thermal-hydro-wind Interconnected System. *Applications Of Computing, Automation and Wireless Systems in Electrical Engineering*, 553(December), 859–863.
- Praseptyo, C., & Pujiyanta, A. (2014). Media Pembelajaran Himpunan Fuzzy Berbasis Multimedia. *Jurnal Sarjana Teknik Informatika*, 2(2), 379–388.
- Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2020). Whale optimization algorithm-based Sugeno fuzzy logic controller for fault ride-through improvement of grid-connected variable speed wind generators. *Engineering Applications of Artificial Intelligence*, 87(October 2019), 103328.
- Qodari, A. (2015). *Informatika dan Kecerdasan Buatan : Kapan Menggunakan MAE dan MSE*.
- Radita, A. (2017). Penalaan Kendali PID untuk Pengendali Proses. *Jurnal Teknologi Elektro, Universitas Mercu Buana*, 8(2), 109–116.
- Roy, A., & Iqbal, K. (2005). PID controller tuning for the first-order-plus-dead-time process model via Hermite-Biehler theorem. *ISA Transactions*, 44(3), 363–378.
- Safitri, M., Triwiyanto, A., & Wahyudi. (2012). Perancangan Sistem Kontrol Genetic-Fuzzy Studi Kasus Pada Pengendalian Top And Side End Point Composition dan Bottom Refluks Temperature Pada Distillation Column. *Transmisi*, 14(3), 85–90.
- Sahu, P. R., Hota, P. K., & Panda, S. (2017). *Whale optimization algorithm for fuzzy lead-lag structure SSSC damping controller design*.
- Setiawan, A., Yanto, B., & Yasdomi, K. (2018). Logika Fuzzy dengan MATLAB (Contoh Kasus Penelitian Penyakit Bayi Dengan Fuzzy Tsukamoto). In *Jayapangus Press* (Issue 13508029).
- Singh, A. K., Tyagi, B., & Kumar, V. (2011). Fuzzy rule-based controller for binary distillation column. *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, ACAI 2011*, 166–169.
- Sitio, S. L. M. (2018). Penerapan Fuzzy Inference System Sugeno untuk Menentukan Jumlah Pembelian Obat (Studi Kasus: Garuda Sentra Medika). *Jurnal Informatika Universitas Pamulang*, 3(2), 104.

- Sivalingam, R., Chinnamuthu, S., & Dash, S. S. (2017). A modified whale optimization algorithm-based adaptive fuzzy logic PID controller for load frequency control of autonomous power generation systems. *Automatika*, 58(4), 410–421.
- Skogestad, S. (1997). *Dynamics And Control Of Distillation Columns - A Tutorial Introduction*. 75(September), 8–10.
- Sutapa, R. B., Setiyono, B., & Wahyudi, W. (2017). Simulasi Model Kendali Kolom Distilasi Biner Wood & Berry Dengan Adaptive Internal Model Control 2 Degree of Freedom (Aimc 2 Dof) Menggunakan Matlab. *Transmisi*, 19(4), 145.
- Sutikno, S., & Waspada, I. (2012). Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani Pada Motor Dc. *Jurnal Masyarakat Informatika*, 2(3).
- Taufiq, G. (2016). Implementasi Logika Fuzzy Tahani untuk Model Sistem Pendukung Keputusan Evaluasi Kinerja Karyawan. *Jurnal Pilar Nusa Mandiri*, 12(1), 12–20.
- Tehrani, K. A., & Mpanda, A. (2012). PID Control Theory. *Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas*, February.
- Vasičkaninová, A., Bakošová, M., & Mészáros, A. (2016). Fuzzy Control of a Distillation Column. *Computer Aided Chemical Engineering*, 38, 1299–1304.
- Wara, B. B., Setiyono, B., & Wahyudi. (2016). Pengontrolan Kolom Distilasi Biner Menggunakan Metode Internal Model Control. *Transient*, 5(2), 166–172.
- Wardani, A. R., Nasution, Y. N., & Amijaya, F. D. T. (2017). Aplikasi Logika Fuzzy Dalam Mengoptimalkan Produksi Minyak Kelapa Sawit Di PT. Waru Kaltim Plantation Menggunakan Metode Mamdani. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, 12(2), 94.
- Wood, R. K., & Berry, M. W. (1973). Terminal composition control of a binary distillation column. *Chemical Engineering Science*, 28(9), 1707–1717.
- Yang, D. R. (2020). *Control of Multi Input Multi Output Processes*.
- Yunita. (2016). *Penerapan Logika Fuzzy dalam Sistem Pendukung Keputusan Pemberian Beasiswa BSM*. XIII(1), 42–49.

LAMPIRAN

A. Hasil *Tuning* Parameter PID

1. Tuning Pengendali Tipe-P Komposisi Atas

Function Block Parameters: PID Controller 1

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **P** Form: **Parallel**

Time domain:

☒ Continuous-time
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Proportional (P): 0.139600056968372 [Compensator formula](#)

Tune... **P**

☒ Enable zero-crossing detection

OK Cancel Help Apply

2. Tuning Pengendali Tipe-P Komposisi Bawah

Function Block Parameters: PID Controller 2

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **P** Form: **Parallel**

Time domain:

☒ Continuous-time
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Proportional (P): -0.0776810998025405 [Compensator formula](#)

Tune... **P**

☒ Enable zero-crossing detection

OK Cancel Help Apply

3. Tuning Pengendali Tipe-PI Komposisi Atas

Function Block Parameters: PID Controller 1

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PI** Form: **Parallel**

Time domain:

☒ Continuous-time
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Proportional (P): [Compensator formula](#)

Integral (I): $P + I \frac{1}{s}$

Tune...

Initial conditions

Source: **internal**

Integrator:

External reset: **none**

OK Cancel Help Apply

4. Tuning Pengendali Tipe-PI Komposisi Bawah

Function Block Parameters: PID Controller 2

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PI** Form: **Parallel**

Time domain:

☒ Continuous-time
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Proportional (P): [Compensator formula](#)

Integral (I): $P + I \frac{1}{s}$

Tune...

Initial conditions

Source: **internal**

Integrator:

External reset: **none**

OK Cancel Help Apply

5. Tuning Pengendali Tipe-PID Komposisi Atas

Function Block Parameters: PID Controller 1

PID Controller
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PID** Form: **Parallel**

Time domain:
☒ Continuous-time
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Proportional (P): 0.334592885111495

Integral (I): 0.0337369935316392

Derivative (D): 0.0576139831698342

Filter coefficient (N): 0.280852400642662

[Compensator formula](#)

$$P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}}$$

Tune...

Initial conditions

Source: internal

OK Cancel Help Apply

6. Tuning Pengendali Tipe-PID Komposisi Bawah

Function Block Parameters: PID Controller 2

PID Controller
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: **PID** Form: **Parallel**

Time domain:
☒ Continuous-time
☐ Discrete-time

Main PID Advanced Data Types State Attributes

Controller parameters

Proportional (P): -0.0270215229633316

Integral (I): -0.012045060685882

Derivative (D): 0

Filter coefficient (N): 100

[Compensator formula](#)

$$P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}}$$

Tune...

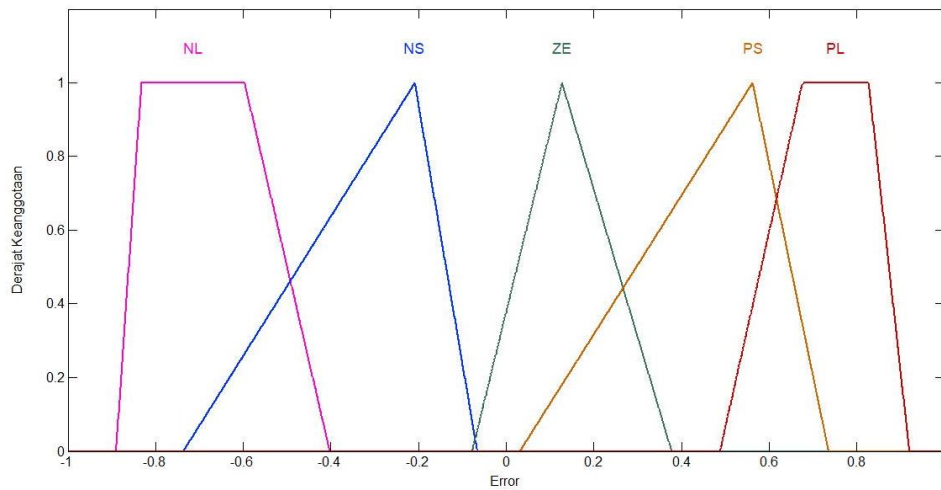
Initial conditions

Source: internal

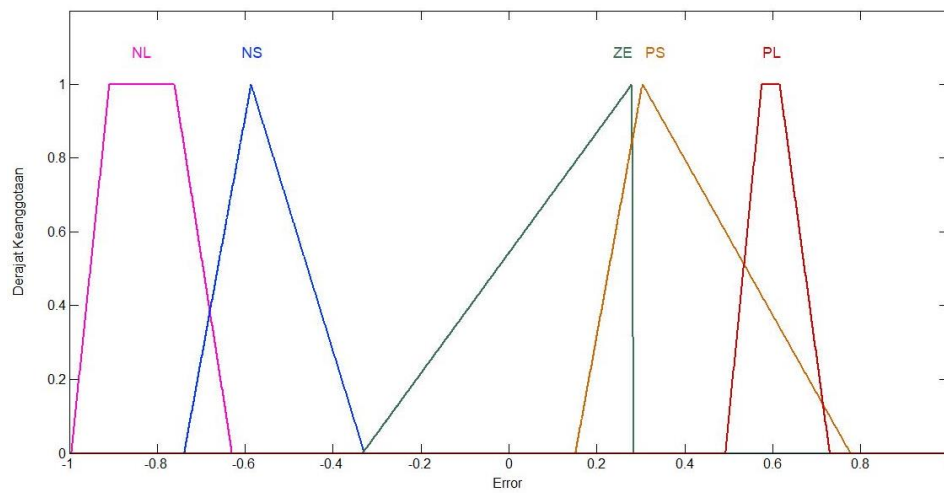
OK Cancel Help Apply

B. Simulasi

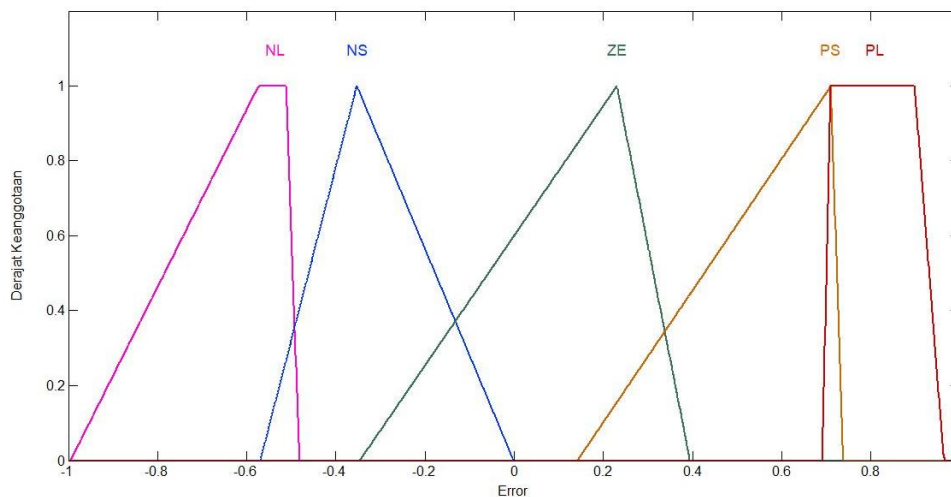
1. Fungsi Keanggotaan *Input Error* Fuzzy-WOA Skenario 2



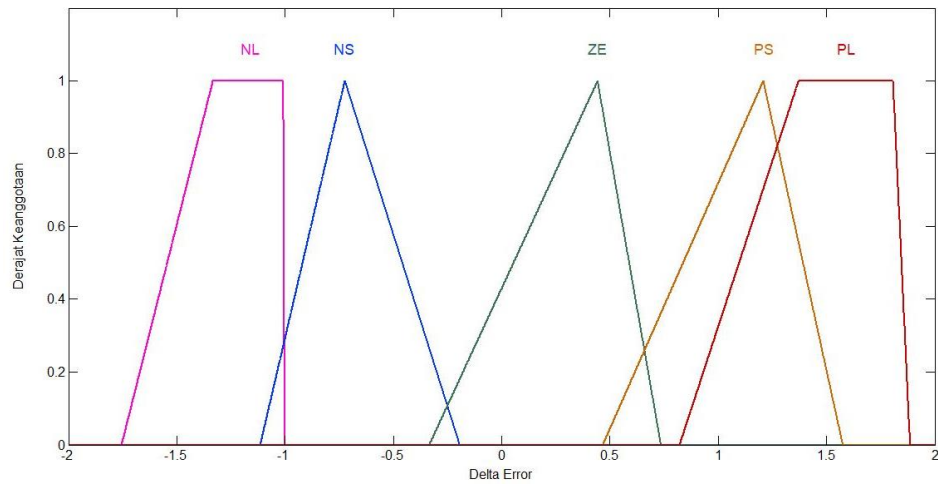
2. Fungsi Keanggotaan *Input Error* Fuzzy-WOA Skenario 3



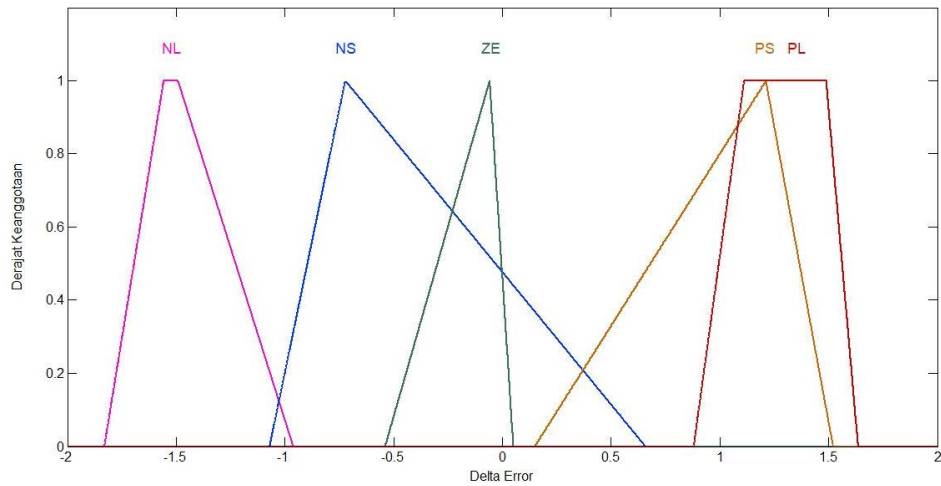
3. Fungsi Keanggotaan *Input Error* Fuzzy-WOA Skenario 4



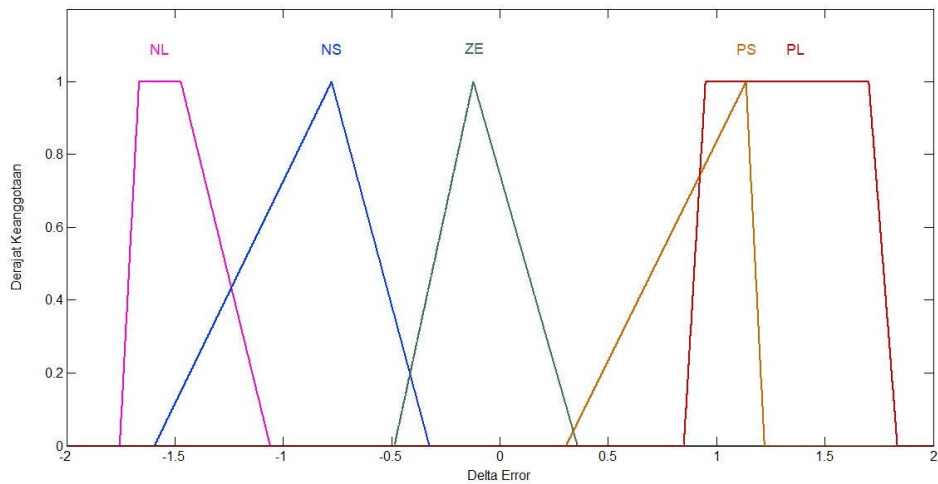
4. Fungsi Keanggotaan *Input Delta Error* Fuzzy-WOA Skenario 2



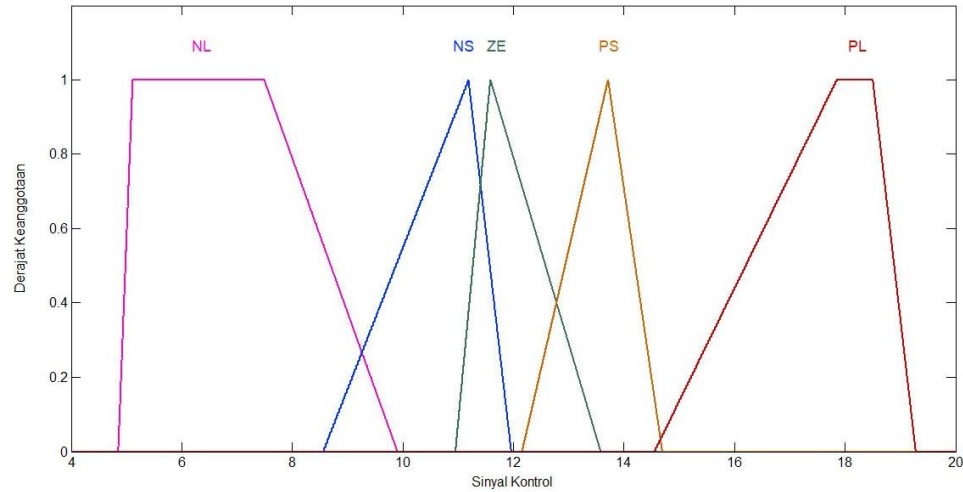
5. Fungsi Keanggotaan *Input Delta Error* Fuzzy-WOA Skenario 3



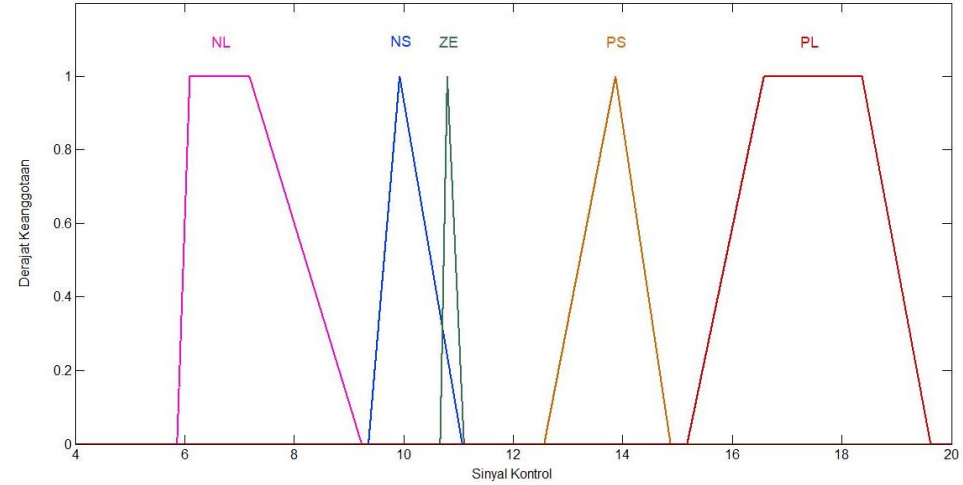
6. Fungsi Keanggotaan *Input Delta Error* Fuzzy-WOA Skenario 4



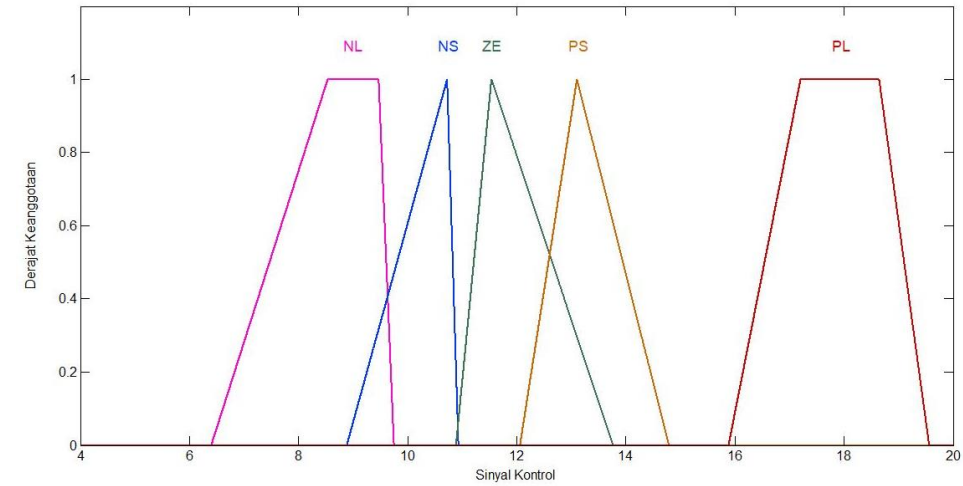
7. Fungsi Keanggotaan *Output* Sinyal kontrol Fuzzy-WOA Skenario 2



8. Fungsi Keanggotaan *Output* Sinyal kontrol Fuzzy-WOA Skenario 3

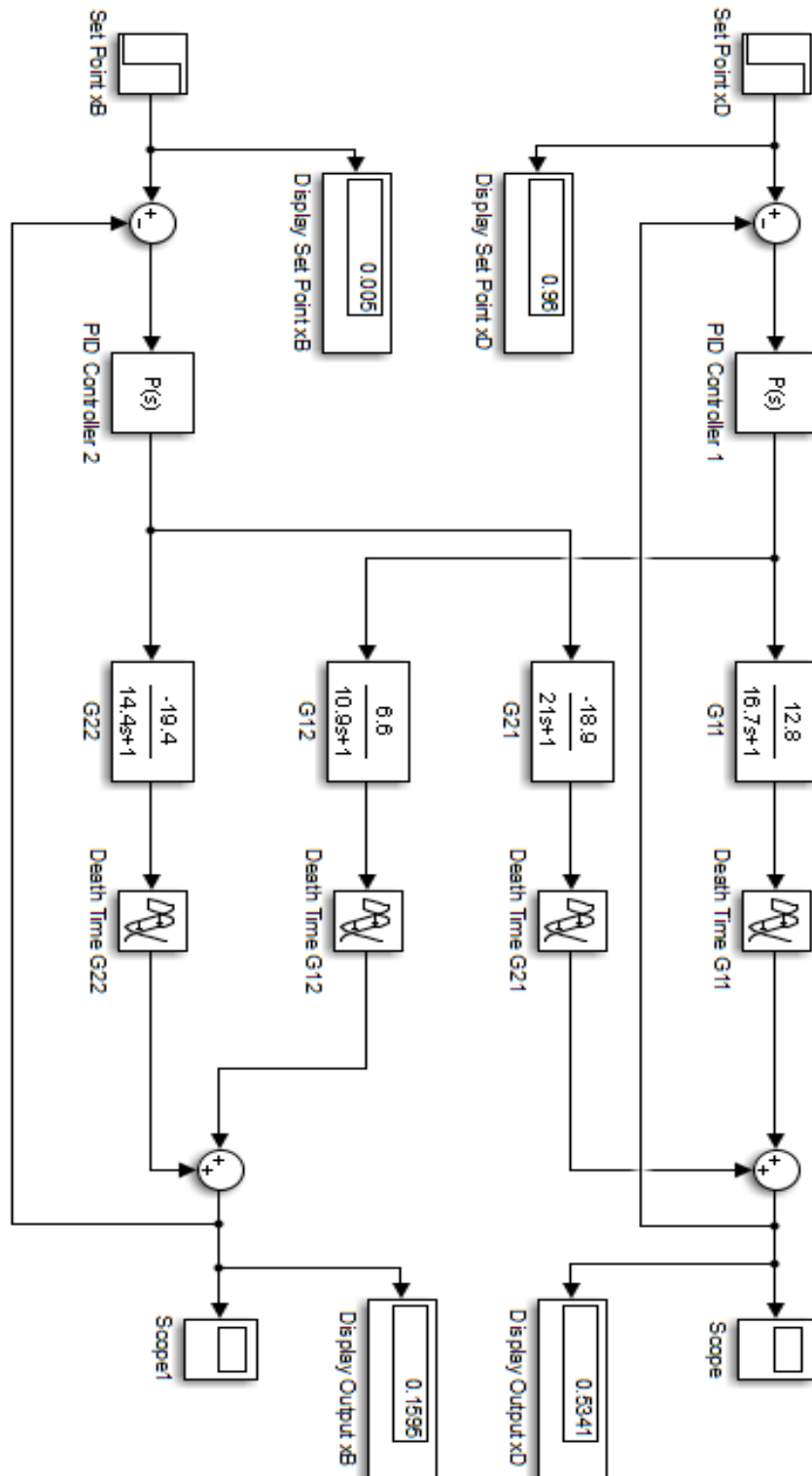


9. Fungsi Keanggotaan *Output* Sinyal kontrol Fuzzy-WOA Skenario 4

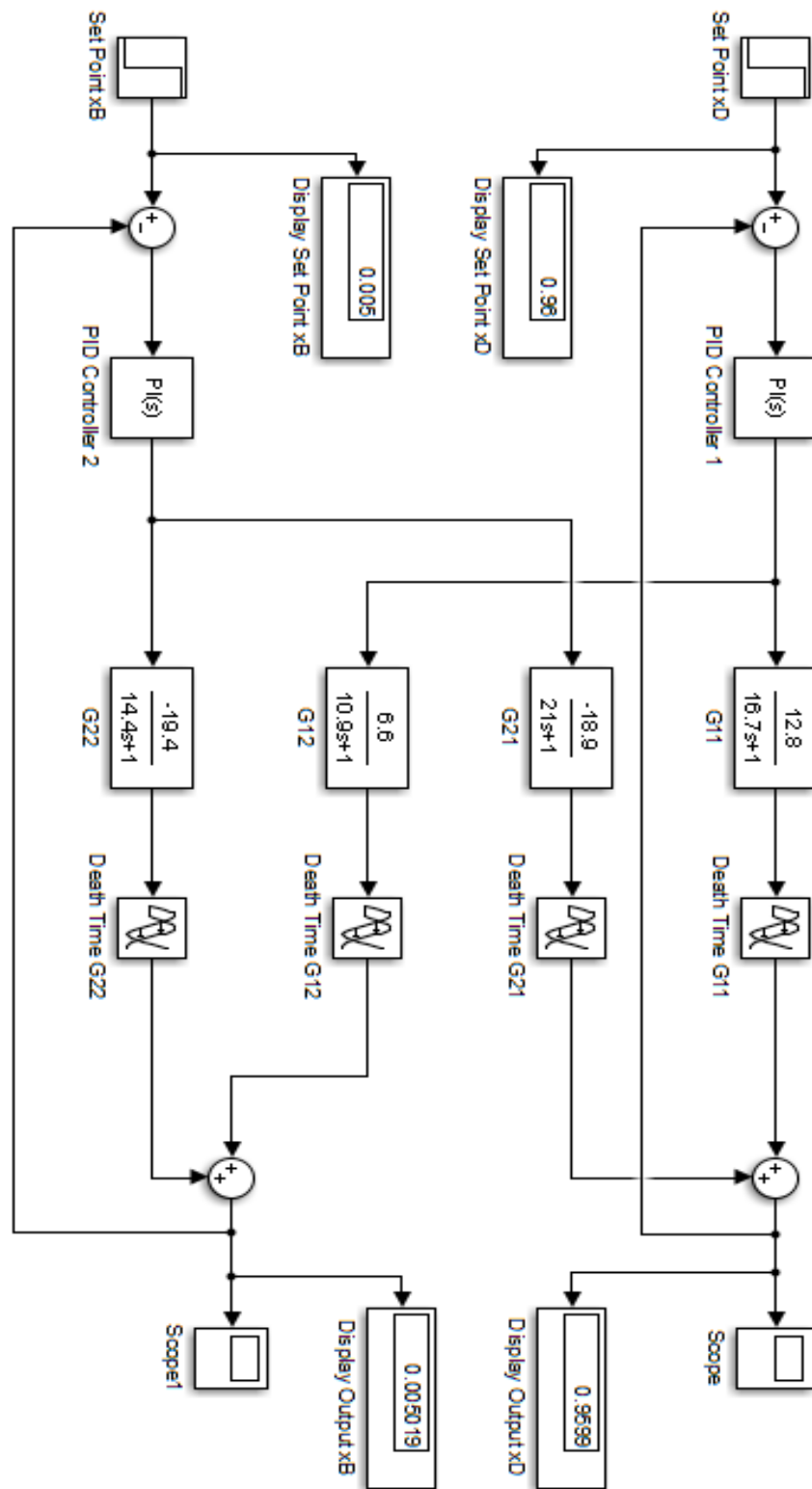


C. Simulasi

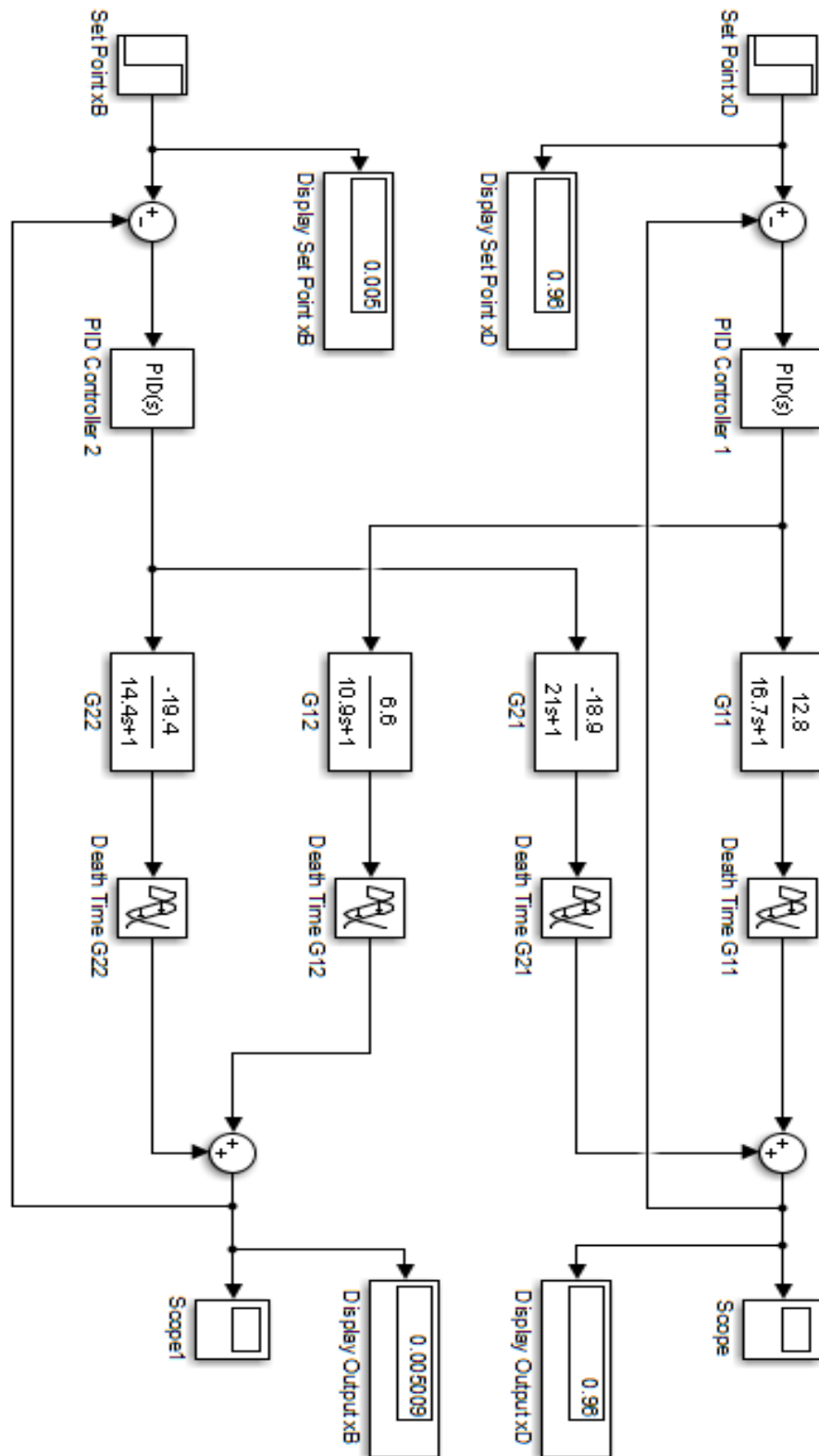
1. Simulasi Pengendali Tipe-P dalam Keadaan *Running*

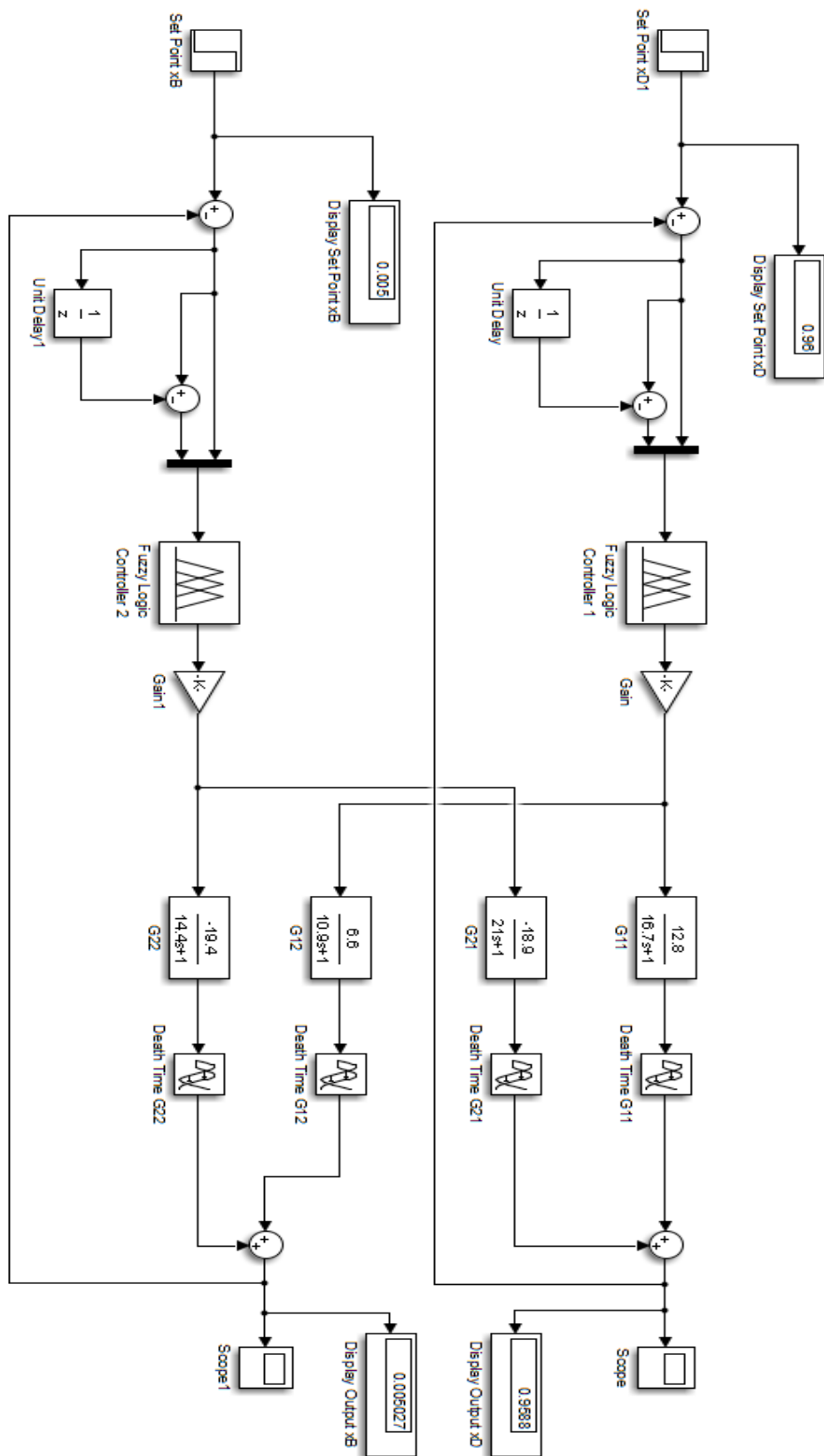


2. Simulasi Pengendali Tipe-PI dalam Keadaan *Running*

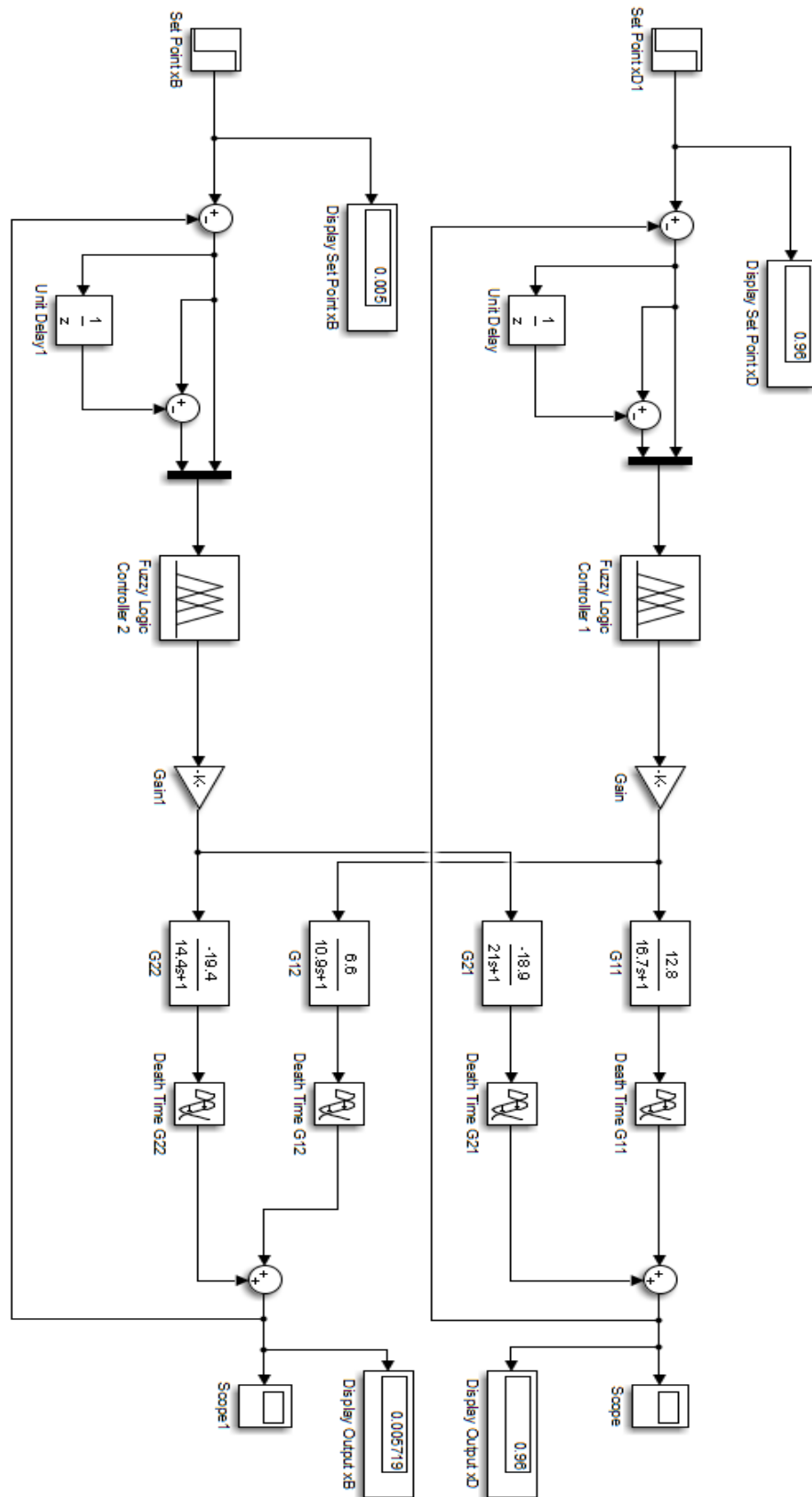


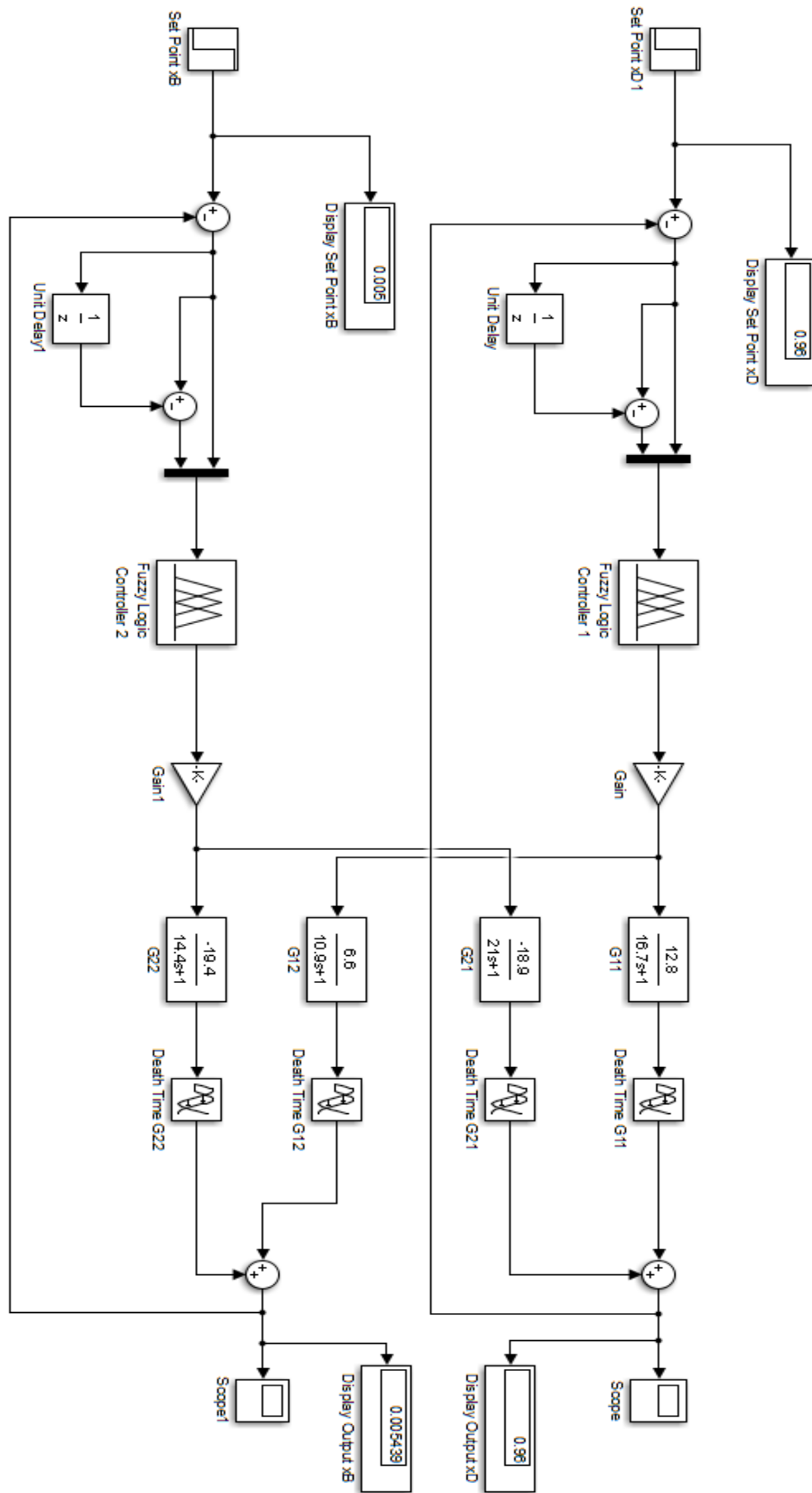
3. Simulasi Pengendali Tipe-PID dalam Keadaan *Running*



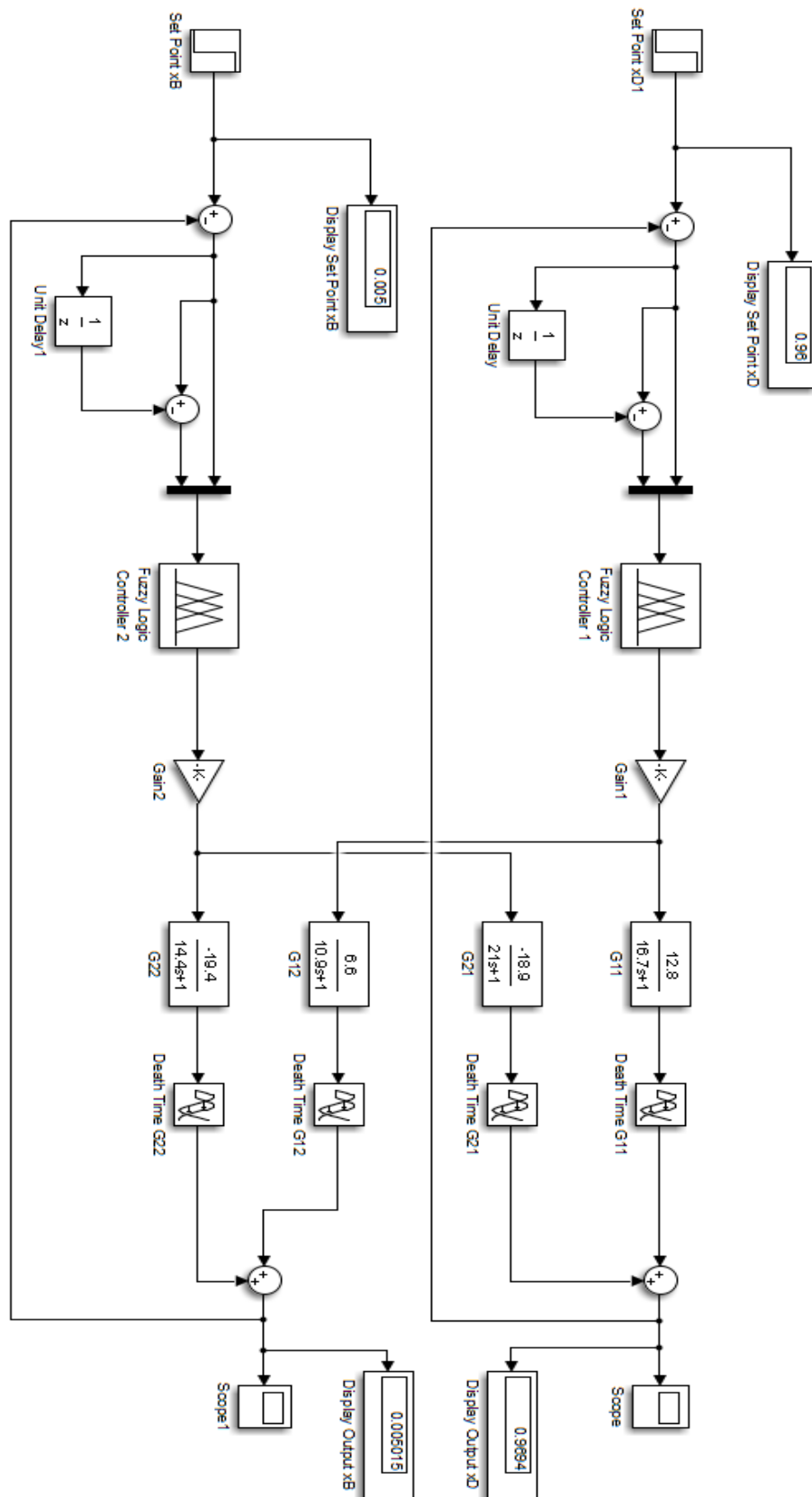
4. Simulasi Pengendali Fuzzy Skenario 1 dalam Keadaan *Running*

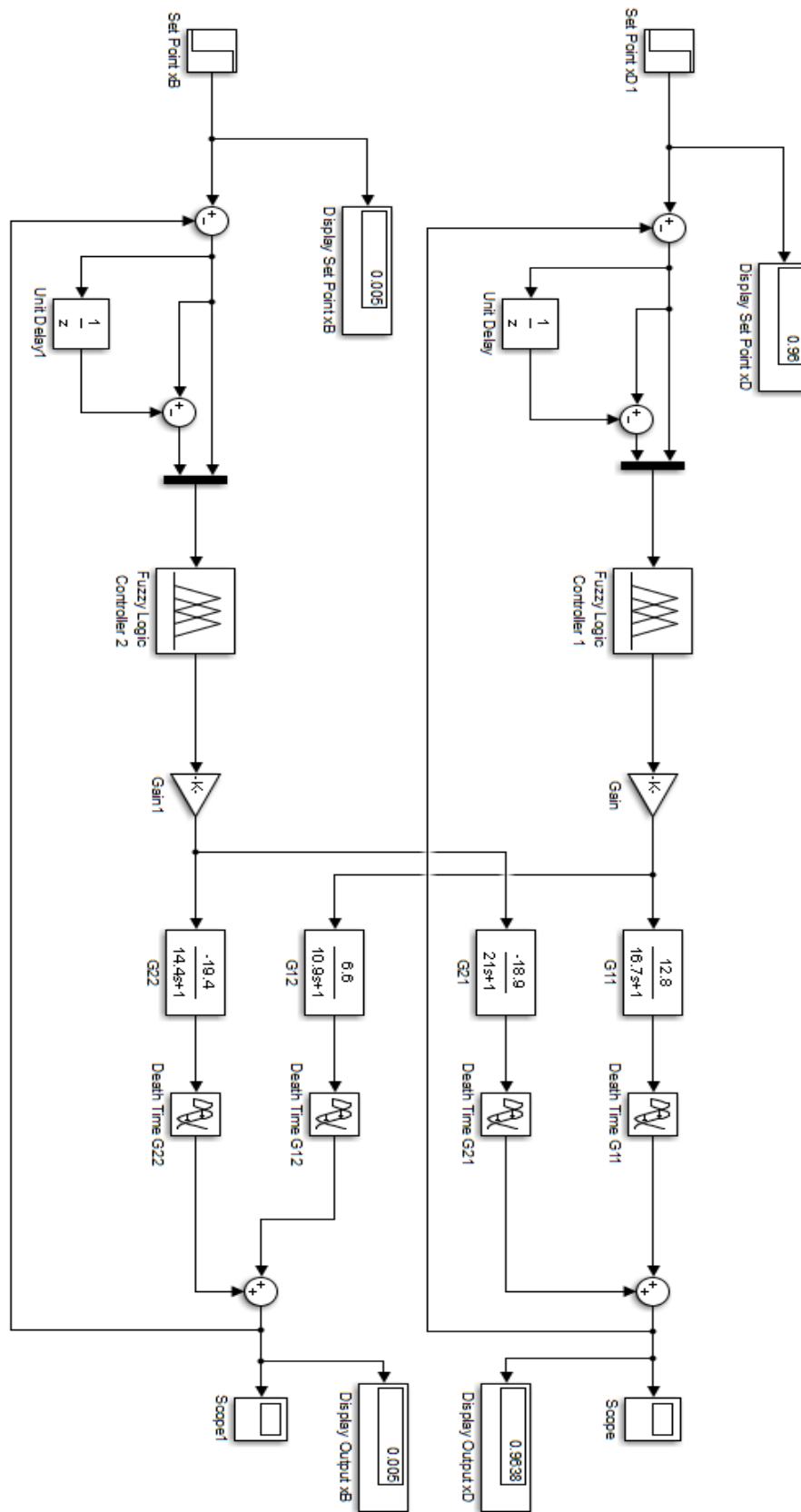
5. Simulasi Pengendali Fuzzy Skenario 2 dalam Keadaan *Running*

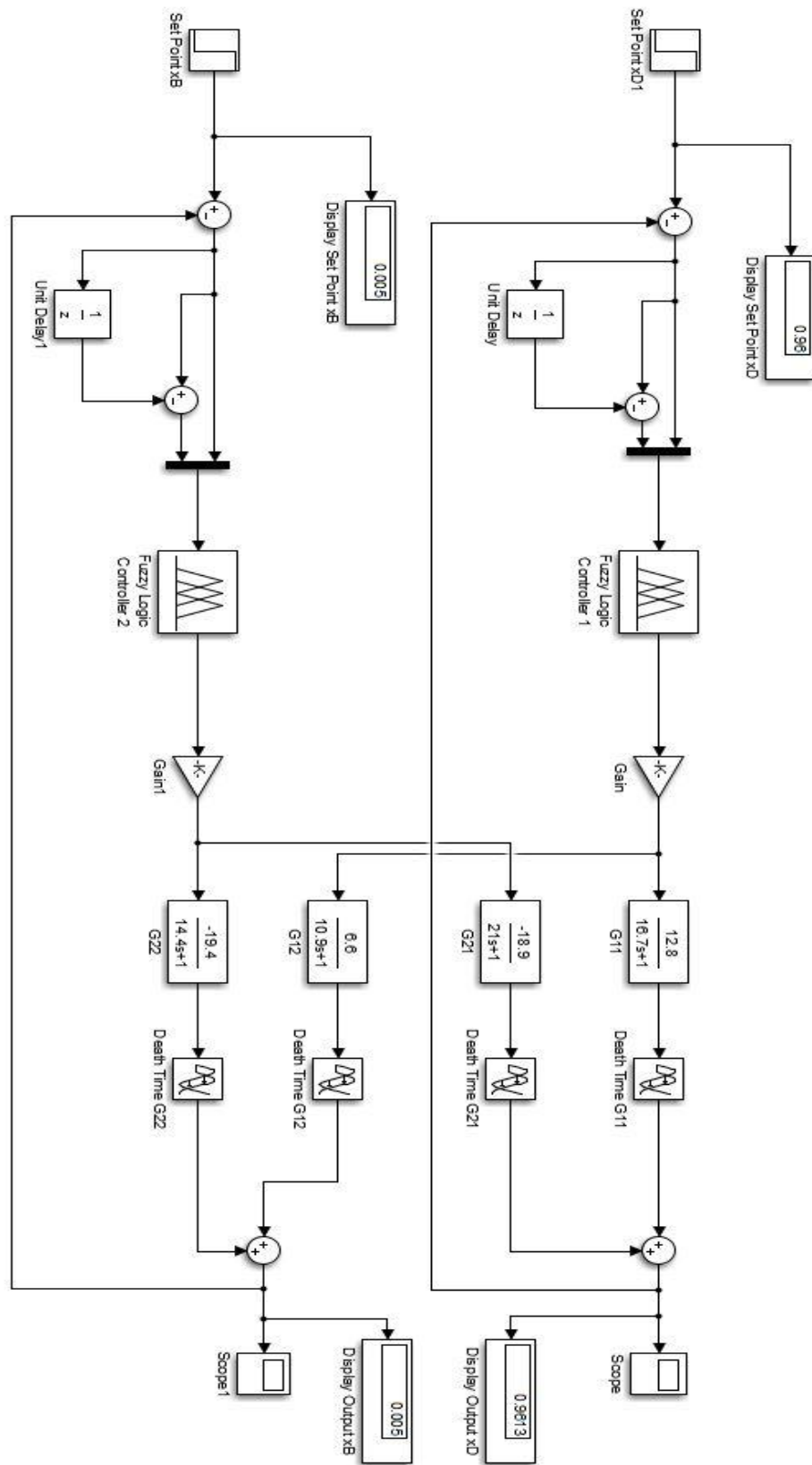


6. Simulasi Pengendali Fuzzy-WOA Skenario 1 dalam Keadaan *Running*

7. Simulasi Pengendali Fuzzy-WOA Skenario 2 dalam Keadaan *Running*



8. Simulasi Pengendali Fuzzy-WOA Skenario 3 dalam Keadaan *Running*

9. Simulasi Pengendali Fuzzy-WOA Skenario 4 dalam Keadaan *Running*

D. Kode Matlab *Whale Optimization Algorithm*

```

clear;
clc;

% Inisialisasi parameter WOA
it_maks = 500;
dim = 4;
lb = 14;
ub = 20;
no_search_agent = 100;

% Untuk looping
t = 0;

% Nilai error untuk disubstitusikan ke fitness
k = FuzzyxD.data;
o = FuzzyxB.data;

% Inisialisasi vektor posisi dan nilai dari leader
Leader_pos = [14 16 20 20];
Leader_score = -inf;
convergence_curve = zeros(1,it_maks);

% Inisialisasi posisi search agent
Boundary_no = size(ub,2); % Jumlah batas
if Boundary_no==1
    Positions=rand(no_search_agent,dim).*(ub-lb)+lb;
end
% Jika nilai lb dan ub berbeda
if Boundary_no>1
    for i=1:dim
        ub_i=ub(i);
        lb_i=lb(i);
        Positions(:,i)=rand(no_search_agent,1).*(ub_i-
lb_i)+lb_i;
    end

```

```

end

% Looping WOA
while t<it_maks
    for i=1:size(Positions,1)

        % Mengembalikan search agent yang melampaui batas
        Flag4ub=Positions(i,:)>ub;
        Flag4lb=Positions(i,:)<lb;

        Positions(i,:)=(Positions(i,:).*(~(Flag4ub+Flag4lb)))+ub.*Flag4ub+
        lb.*Flag4lb;

        % Hitung fitness setiap search agent dengan MSE
        fitness = (sum(k.^2))/length(k)+(sum(o.^2))/length(o);

        % Update leader
        if fitness > Leader_score
            Leader_score = fitness; % Update alpha
            Leader_pos = Positions(i,:);
        end
    end

end

% a berkurang secara linier dari 2 ke 0
a=2-t*((2)/it_maks);

% a2 berkurang secara linier dari -1 ke -2 untuk
menghitung t
a2=-1+t*((-1)/it_maks);

% Update posisi search agents
for i=1:size(Positions,1)
    r1=rand(); % r1 = random number pada [0,1]
    r2=rand(); % r2 = random number pada [0,1]

    A=2*a*r1-a; % pers (2.41)

```

```

C=2*r2;          % pers (2.42)

b=1;              % parameter pada pers (2.43)
l=(a2-1)*rand+1;  % parameter pada pers (2.43)

p = rand();

for j=1:size(Positions,2)

    if p<0.5
        if abs(A)>=1

            % Fase eksplorasi (Search for Prey)
            rand_leader_index =
floor(no_search_agent*rand()+1);
            X_rand = Positions(rand_leader_index, :);
            D_X_rand=abs(C*X_rand(j)-Positions(i,j));
% Pers (2.47)

            Positions(i,j)=X_rand(j)-A*D_X_rand;
% Pers (2.48)

        elseif abs(A)<1

            % Encircling Prey
            D_LLeader=abs(C*Leader_pos(j)-
Positions(i,j)); % Pers (2.39)
            Positions(i,j)=Leader_pos(j)-A*D_LLeader;
% Pers (2.40)

        end

    elseif p>=0.5

        % Bubble-net attacking
        distance2Leader=abs(Leader_pos(j)-
Positions(i,j)); % Pers (2.44)

```

```
Positions(i,j)=distance2Leader*exp(b.*l).*cos(l.*2*pi)+Leader_pos(j); % Pers (2.43)

    end

end

end
t=t+1;
convergence_curve(t)=Leader_score;
hasil = [t Leader_score Leader_pos]
end
```

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Agam Prihatna merupakan nama lengkap dari penulis, dengan nama panggilan Agam. Penulis dilahirkan di Kota Pamekasan, pada tanggal 14 Agustus 1997. Saat ini, penulis tinggal di Jalan P. Trunojoyo V/15, Pamekasan. Penulis merupakan anak pertama dari tiga bersaudara. Riwayat pendidikan penulis adalah: SDI Al-Munawwarah (2004-2010), SMPN 1 Pamekasan (2010-2013), SMAN 1 Pamekasan (2013-2016), dan mengawali perkuliahan di Departemen Teknik Fisika ITS pada tahun 2016. Selama perkuliahan, penulis aktif sebagai asisten Laboratorium Pengukuran, Keandalan, Risiko, dan Keselamatan. Penulis juga aktif dalam bidang non-akademik, diantaranya menjadi anggota aktif PSM ITS, menjadi staff ITS Expo, menjadi pembawa acara di beberapa *event* PSM ITS, dan lain sebagainya. Hobi yang dimiliki penulis adalah bermain *game*, menyanyi, *beracting*, memasak, jalan-jalan, bermain *voice over*, bermain alat musik, berkuliner, dan menonton video lucu.

Pada Bulan Juli 2020 penulis telah menyelesaikan tugas akhir dengan judul **“Analisis Performansi Sistem Pengendalian pada Kolom Distilasi Menggunakan Fuzzy-Whale Optimization Algorithm (Fuzzy-WOA)”**. Bagi para pembaca yang memiliki kritik, saran, atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini, maka dapat menghubungi penulis melalui *email*: agamprihatna@outlook.com